# POLITECNICO DI TORINO Repository ISTITUZIONALE

## Event-Driven Encoding Algorithms for Synchronous Front-End Sensors in Robotic Platforms

Original

Event-Driven Encoding Algorithms for Synchronous Front-End Sensors in Robotic Platforms / Ros, P. M.; Laterza, M.; Demarchi, D.; Martina, M.; Bartolozzi, C.. - In: IEEE SENSORS JOURNAL. - ISSN 1530-437X. - ELETTRONICO. - 19:16(2019), pp. 7149-7161. [10.1109/JSEN.2019.2911668]

Availability: This version is available at: 11583/2755952 since: 2021-11-08T17:01:32Z

*Publisher:* Institute of Electrical and Electronics Engineers Inc.

Published DOI:10.1109/JSEN.2019.2911668

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright IEEE postprint/Author's Accepted Manuscript

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

# Event-Driven Encoding Algorithms for Synchronous Front-End Sensors in Robotic Platforms

Paolo Motto Ros<sup>2\*</sup>, Member, IEEE, Marino Laterza<sup>1\*</sup>, Danilo Demarchi<sup>3</sup>, Senior Member, IEEE, Maurizio Martina<sup>3</sup>, Senior Member, IEEE, and Chiara Bartolozzi<sup>1</sup>, Member, IEEE

Abstract—Asynchronous, event-driven, sampling techniques 1 adapt the sampling rate of sensory signals to their dynamics, 2 by effectively compressing the data with respect to synchronous, 3 clock-driven, sampling. In robotics such techniques offer data and 4 bandwidth reduction, together with high temporal resolution and 5 low latency. Despite vision and auditory event-driven sensors are 6 currently available, robots are still equipped with a plethora of other sensors that might benefit from the event-driven encoding. 8 In this paper, we study five estimation algorithms that implement 9 event-driven encoding for off-the-shelf clock-driven sensors. Dig-10 ital accelerometer datasets were used to validate the system in 11 robotic applications; other datasets have been used to assess the 12 general performance of the proposed approach. The two best 13 algorithms in terms of six performance parameters have been 14 implemented on a Xilinx Artix-7 FPGA platform, using 2892 15 LUTs and 3620 flip-flops and reducing the output bandwidth 16 from -44 % to -75 %, over the considered datasets. 17

Index Terms—Asynchronous sampling algorithms, Event-18 Driven, FPGA, Relative Threshold, Output Bandwidth, Robotic 19 **Environment.** 20

21 22

#### I. INTRODUCTION

State-of-the-art sensors are mostly based on clock-driven 23 sampling of the physical signal being measured. This approach 24 has a trade-off between the amount of data acquired and 25 the maximum detectable input frequency (Nyquist) of the 26 signal variation. Tuning the clock-rate for very fast signals 27 results in sampling redundant values when the signal is slowly 28 changing, while decreasing the sampling frequency results in 29 missing potentially important signal variations. Additionally, 30 for sensors with multiple sensing sites, such as cameras or 31 large area tactile devices, there is an inherent latency in data 32 transmission, due to the need to synchronously sample all 33 the sensing elements in the device. While the advantage of 34 clock-driven sampling is the compliance of all sensors and 35 acquisition devices to the clock-driven paradigm, the trade-offs 36 and downsides listed above are detrimental for building effi-37 cient sensory systems for artificial devices. Specifically, data 38 compression, high temporal resolution and short latency are 39

M. Laterza and P. Motto Ros equally contributed to this manuscript

<sup>1</sup> M. Laterza and C. Bartolozzi are with iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy, E-mail: marino.laterza@iit.it, chiara.bartolozzi@iit.it

<sup>2</sup> P. Motto Ros is with Electronic Design Lab, Istituto Italiano di Tecnologia, Genova, Italy, E-mail: paolo.mottoros@iit.it

M. Martina and D. Demarchi are with Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Torino, Italy, E-mail: maurizio.martina@polito.it, danilo.demarchi@polito.it

especially desirable in robotics, where the progressive appli-40 cation in unconstrained scenarios is leading to the integration 41 of an increasing number of sensors needed to perceive both the 42 environment and the status of the robot. A viable alternative is 43 the "neuromorphic" event-driven sensing strategy inspired by 44 biological sensory systems, where the sensing element (e.g. 45 photoreceptors and retinal cells in vision, mechanoreceptors 46 in touch) gets active when it detects a variation in its own 47 input and sends action potentials to neurons in the sensing 48 areas of the brain. The output activity of each sensing neuron 49 encodes for the properties of the sensed stimulus. Similar 50 approaches (eventually sending a sample along with the event) 51 have been investigated and developed in other research areas 52 too, including (but not limited to) automation control and 53 signal processing [4] and energy metering [5]. 54

Neuromorphic event-driven sensing sends data only when the amplitude of the measured signal has experimented a certain change, rather than at fixed time intervals. This change 57 could be referred to the sample which generated the last event [1]. The signal is assumed to stay constant until another 59 event is produced. As a consequence, the reference value could be exploited as a predictor for the future values of the signal. If this estimate differs from the actual sample value by more than a given amount, then a new event is generated. In this encoding scheme, the data is written on the output bus as soon as the change is sensed. In a scenario with many sensing sites (e.g., vision or tactile), this strategy decreases latency dramatically, avoiding the sampling and transmission of the whole set of pixels (or taxels). Information is then encoded in the relative timing between generated events and the value of the sample is not sent, limiting the number of bits to be sent, as opposed to other asynchronous sampling transmissions, which send the whole data sample whenever a threshold-crossing occurs [2]-[4].

This approach resulted so far in the design of event-driven 74 vision [8]-[10], [12], auditory [11] and (more recently) tactile 75 sensing [13], [18], [22], where the sensing element itself im-76 plements the data-driven sampling. While event-driven vision 77 sensors have already been integrated on robotic platforms, 78 tactile sensors require further development [21] and other 79 sensor modalities are not yet under development. On the other 80 hand, robots are fully equipped with a plethora of sensors 81 (temperature, pressure, encoders, accelerometers, etc.) and it is 82 possible to emulate event-driven compression using traditional 83 off-the-shelf clock-driven sensors that are readily integrated 84 in robotic environments. The aim of this approach is two-85 fold: improving efficiency in signal transmission (optimizing 86

55

56

58

60

61

62

63

64

65

66

67

68

69

70

71

72

bandwidth, data compression, latency, etc.) and delivering 87 prototype systems for the development of event-driven algo-88 rithms for perception. An example of emulation of event-89 driven sensing of clock-based data has been shown in [20], 90 where the event-driven data encoding and transmission have 91 been implemented on an FPGA interfaced to the capacitive 92 sensors of the iCub robot skin [24]. Other applications have 93 been developed in the prosthetic field [14], with the aim of 94 implementing a tactile feedback control, based on the actual 95 data sensed by the prosthesis. 96

<sup>97</sup> Algorithms proposed for the conversion of clock-sampled <sup>98</sup> data into event-driven show the potential compression perfor-<sup>99</sup> mance of this approach. They are based on detection of relative <sup>100</sup> change among current and previous samples. Specifically, the <sup>101</sup> detected change is relative to the absolute value ( $\Delta x/x$ ), de <sup>102</sup> facto implementing a logarithmic compression that increases <sup>103</sup> the compression dynamic range.

In this work, we characterize a set of more complex algorithms, analyzing their performance in terms of data compression and implementation cost. Our goal is to find an algorithm for event-driven encoding that can be applied to any sensory signal acquired by a clock-sampling strategy.

As case study, we tested the algorithms for the encoding 109 of MEMS accelerometers that are integrated in the iCub 110 humanoid robotic platform [24]. As discussed in [20], the mid-111 term goal is to have a unified tactile/accelerometric sensing 112 system in order to enable the development of event-driven 113 applications - allowing the humanoid robot to interact with 114 the surrounding environment - without requiring the devel-115 opment of new sensors. With this aim, one of the requirement 116 has been to use the same hardware platform (and to respect 117 the same implementation constraints) as done in [20]. 118

The conceived scheme is absolutely general and flexible, so that changing its internal parameters will produce good performance for very different sensors. To offer a thorough analysis of the encoding scheme, we frame it as an estimation problem and compare different solutions.

Starting from the asynchronous algorithm called "Send-124 on-Delta" [1], where the sample is transmitted when the 125 absolute value of the difference between the current input 126 and the previous one is greater than a given threshold, we 127 also evaluated more complex algorithms. Those algorithms 128 compare the input data with a reference value, in order to 129 decide if an event has to be generated or not. As such, the 130 reference value could be thought of as a predictor of the value 131 of the next sample. For example, in the Send-on-Delta case, the 132 predictor is a zero-order one. As a result, in this alternative 133 view, if the estimation error falls within a given boundary 134 with respect to the measured input, no event (and hence no 135 transmission) is generated. Differently from the standard Send-136 on-Delta, in the proposed implementation, as soon as the 137 estimation error exceeds the boundary, an event is generated 138 and transmitted. The information is encoded in the exact time 139 at which the event is generated and implicitly transmitted in 140 the timing between events, hence, we do not need to send the 141 absolute value of the sample together with the event. In order 142 to bound the maximum relative error, the change with respect 143 to the predicted sample value is computed using a relative 144

threshold.

The best-performing algorithms are then evaluated in terms 146 of accuracy, output data rate reduction and resource re-147 quirement on a Xilinx Artix-7 FPGA (model XC7A35T-L1). 148 Specifically, the accuracy parameter (the error between the 149 original and the reconstructed signal) is used to check if and 150 how much the proposed encoding reduces the information 151 content of the signal, however, in neuromorphic perception 152 the signal is not usually reconstructed and information about 153 the sensed signal is extracted using event-driven algorithms. 154 Also, the compression rate is used to compare the different 155 proposed algorithms, rather than to find the best possible 156 compression strategy for the signal at hand. For this reason, we 157 treated agnostically the signal that we were processing, without 158 using knowledge about the physical origin of the signal 159 itself. That strategy could benefit a specific application, for 160 example by considering the non-independence among the three 161 axis of the accelerometers and the relationship between the 162 position, velocity and acceleration values. In such a case, the 163 compression would be higher, but specific to the accelerometer 164 signal only. Rather, we were looking for a more general 165 algorithm for the event-driven encoding of any sensory signal. 166 In the accelerometer case, the architecture consists in three 167 identical submodules, one per spatial axis, which implement 168 the encoding scheme in parallel. The approach used to send the 169 data on the bus is the so-called Address Event Representation 170 (AER) [15]–[17], where the output data only includes the event 171 polarity (e.g., if the signal is increasing or decreasing with 172 respect to the previous sample) along with the corresponding 173 address of the sensor (in this specific case, of the accelerometer 174 axis). Consequently, the output bus exhibits an asynchronous 175 flow of messages containing the sender address and some 176 bits representing the event polarity information, instead of 177 the whole sample value. As specific AER protocol, we use 178 an implementation of the asynchronous serial AER [15]-179 [17], that is specifically designed for robotic systems, where 180 minimizing wiring is a strict requirement. We first outline 181 the main differences between the traditional asynchronous 182 sampling schemes and the proposed one (Section II). After 183 setting up the main features of the implemented scheme, the 184 possible event polarities are discussed (Section III). Once 185 the communication system is set, we make a comparison 186 aimed at identifying the best predicting algorithm to embed 187 inside the defined scheme (Section IV). Then, we evaluate 188 the quantitative results of this analysis by means of several 189 performance figures (Section V) and we implement the best-190 performing algorithms, as well as the communication system, 191 on FPGA (Section VI). The main achievements of the work 192 and future development are finally discussed (Section VII). 193

#### II. EVENT-GENERATION SCHEME

#### A. Pre-Algorithmic Manipulation of Data

The sampling rate of most commercial digital accelerometers ranges from units to thousands of Hz. It is suitable for a clock-based acquisition from this category of sensors, but an event-driven transmission requires a higher temporal precision. Specifically, the information in event-driven transmission is

145



Fig. 1: Overall scheme of the proposed system: the analog signal  $x_{TX}(t)$  is first acquired by the sensor and then resampled at a higher  $1/T_s$  rate to improve the timing accuracy; by comparing the state variable  $x_{TX}^*$  with the interpolated  $x'_{TX}$ , events are eventually generated (EvGen); on the decoding side, a corresponding state variable  $x_{RX}^*$  is combined with the reception of events in order to update the output variable  $x'_{RX}$ .

triggered exactly by change detection, whereas in clock-based
 sampling this situation can happen between two consecutive
 samples.

In the proposed implementation, we internally resample, 204 with a constant period  $T_s$ , the data received from the trans-205 ducer x[k] that corresponds to x(kT) (in the discrete time 206 domain, regularly sampled with a period T) using a linear in-207 terpolator and resulting in  $\tilde{x}[i]$  that corresponds to  $\tilde{x}(iT_s)$  (with 208 the only constraint  $\tilde{x}(kT) = x(kT)$ , and, usually,  $T_s \ll T$ ). 209 The resampling rate has been fixed at 5 MHz, as in [20], 210 211 leading to a flow of produced samples every 200 ns. The denser input mimics an analog continuous signal and allows to timely 212 detect the variation of the input signal, therefore increasing 213 the timing precision of the overall acquisition system. The 214 increased timing precision allows for a better reconstruction 215 of the signal, limiting the loss of information that would oc-216 cur without signal interpolation. The corresponding hardware 217 block will contain a resampling unit, or resampler, feeding the 218 block implementing the event generation (EG). 219

## 220 B. Algorithm Scheme

Figure 1 shows the overall scheme of an event transmitter 221 (TX), including the event generation, and the receiver (RX), 222 including the sample reconstruction. The event generation 223 block computes the estimate for the sample at the next time 224 step — by using one of the algorithms detailed in Section III 225 - and compares it to the sample produced by the resampler. 226 When the absolute value of their difference exceeds a given 227 threshold, an event is generated. Namely, for each point of 228 the linear interpolation, a predicted value is computed: the 229 estimation always depends on the type of the last sent event 230 and it may further depend either on the last value(s) which 231 produced event(s) or on the last estimate(s). 232

In time-continuous asynchronous sampling algorithms [1]– [4], [6], sampling (and therefore event generation, EvGen) occurs whenever the absolute difference between the predicted value and the input signal (sample) crosses a given (static) threshold, i.e., we can define the sequence  $\mathcal{T} \triangleq \{t_i\}$  (with  $t_i \in \mathbb{R}^+$  and  $i \in \mathbb{N}$ ) as:

$$\mathcal{T} \triangleq \{t_i \mid |x_{TX}^{\Delta}(t_i)| \ge \delta(t_i) \land |x_{TX}^{\Delta}(t_i^-)| < \delta(t_i^-)\}$$
(1)

where  $\wedge$  refers to the logical AND operation, and we define

$$x_{TX}^{\Delta}(t) \triangleq x_{TX}(t) - \hat{x}_{TX}(t)$$
$$\delta(t) \triangleq \delta^*$$
(2)

with  $\hat{x}_{TX}(t)$  the predicted value at t,  $\delta(t)$  the threshold set to a fixed value  $\delta^*$ . Since we are dealing with samplebased sensors, with the output data eventually re-sampled or interpolated (at a fixed rate, as described in Sec. II-A), we can define a new sequence  $\mathcal{I} \triangleq \{i\}$ , corresponding to the sequence  $\{t_i = iT_s\}$  (with  $i \in \mathbb{N}$ ) of discrete time-points at which an event is generated, as follows:

$$\mathcal{I} \triangleq \{i \mid |x_{TX}^{\Delta}[i]| \ge \delta[i] \land |x_{TX}^{\Delta}[i-1]| < \delta[i-1]\}$$
(3)

where we define

$$\begin{aligned} x_{TX}^{\Delta}[i] &\triangleq \tilde{x}_{TX}[i] - \hat{x}_{TX}[i] \\ \delta[i] &\triangleq \delta^* \end{aligned}$$
(4)

being  $\tilde{x}_{TX}[i]$  and  $\hat{x}_{TX}[i]$  the interpolated and predicted, respectively, signals,  $\delta[i]$  the threshold set to a fixed (positive) value  $\delta^*$ .

With this scheme, two events are sufficient to implement 249 an unambiguous communication. We can define a sequence of events  $\mathcal{E} \triangleq \{E_i\}$ , with each  $E_i$  triggered at time  $t_i \in \mathcal{T}$  (in 251 case of time-continuous system) or, with the resampling (as 252 in this case), at  $i \in \mathcal{I}$ , as: 253

$$E_{i} = \begin{cases} E_{\uparrow} & \text{if } x_{TX}^{\Delta}[i] \ge \delta[i] \\ E_{\downarrow} & \text{if } x_{TX}^{\Delta}[i] \le -\delta[i]. \end{cases}$$
(5)

 $E_{\uparrow}$  and  $E_{\downarrow}$  are usually encoded by a polarity bit added to the address of the event [8], [16], [17]. Events are transmitted from the TX to the RX in real-time with a low-latency/low-overhead point-to-point asynchronous AER protocol [19], [20], so that events can be immediately processed by the RX.

On the RX side, whenever an event is received at time  $^{259}$ t (EvRcv(t)), the corresponding  $\tilde{x}_{RX}(t)$  can be updated as  $^{260}$ follows:  $^{261}$ 

$$\tilde{x}_{RX}(t) = \begin{cases} \hat{x}_{RX}(t^-) + \delta(t) & \text{if } \operatorname{EvRcv}(t) = E_{\uparrow} \\ \hat{x}_{RX}(t^-) - \delta(t) & \text{if } \operatorname{EvRcv}(t) = E_{\downarrow} \\ \hat{x}_{RX}(t^-) & \text{otherwise} \end{cases}$$
(6)

where  $\hat{x}_{RX}(t)$  is the predicted value at time t by the RX. This scheme, even if based simply on a difference, is not

263 optimal in the application under investigation. Indeed, the 264 accelerations provided by the sensor could go as high as 4G 265 and as low as -4 G, crossing the zero level, as proved in several 266 datasets acquired on the iCub robotic equipment [24]. For 267 an equal relative change (eg. 10% of the reference value), 268 a fixed threshold produces more events for a higher sensor 269 value. However, using fixed thresholds the sensitivity of the 270 event generator is constant for the whole range of the sensor. 271 The advantage of a relative threshold is that the produced 272 number of events is constant for equal relative changes, but 273 the sensitivity decreases for higher sensor values. 274

In order to obtain an homogeneous event rate over the whole range and to compare several algorithms for equal maximum relative error, a relative-threshold-based scheme has been used, instead. A fixed relative threshold is equivalent to a dynamic absolute one, so in the mathematical model described by equations (3) and (4) we can replace equation (4) with:

$$\delta[i] \triangleq \mu \cdot |\tilde{x}_{TX}[i]| \tag{7}$$

where  $\mu$  is the relative threshold, in the range (0, 1), and is a constant time-invariant tuning parameter, which is set once. As a consequence, the relative threshold  $\mu$  describes how the sensitivity of the event generator increases for small sensor values and how the sensitivity decreases for high sensor values. By substituting equation 7 in 3 we can rewrite 3 as:

$$\mathcal{I} \triangleq \{i \mid |x_{TX}^{\delta}[i]| \ge \mu \land |x_{TX}^{\delta}[i-1]| < \mu\}$$
(8)

<sup>287</sup> where we define

$$x_{TX}^{\delta}[i] \triangleq \frac{\tilde{x}_{TX}[i] - \hat{x}_{TX}[i]}{\tilde{x}_{TX}[i]} \tag{9}$$

so to highlight the relative behavior (w.r.t.  $\tilde{x}_{TX}[i]$ ) of the event generation.

To summarize, an  $E_{\uparrow}$  event with a relative threshold  $\mu$ 290 means that the interpolated value is, in absolute value, 100µ % 291 (or more) higher than the predicted one, whereas an  $E_{\downarrow}$  event 292 means that the absolute value of the interpolated sample is 293  $100\mu\%$  (or more) lower than the corresponding predicted 294 one. However, thanks to the resampling mechanism, an exact 295 100µ% difference can be expected. This shall be confirmed 296 by the software simulations on the considered input datasets, 297 which are discussed in Section V. 298

As a result, the reconstructed signal at the receiver  $\tilde{x}_{RX}(t)$ can be obtained as follows:

$$\tilde{x}_{RX}(t) = \begin{cases} \hat{x}_{RX}(t^{-}) \cdot (1+\mu) & \text{if } \operatorname{EvRcv}(t) = E_{\uparrow} \\ \hat{x}_{RX}(t^{-}) \cdot (1-\mu) & \text{if } \operatorname{EvRcv}(t) = E_{\downarrow} \\ \hat{x}_{RX}(t^{-}) & \text{otherwise} \end{cases}$$
(10)

where  $\hat{x}_{RX}(t)$  is the predicted value at time t by the RX.

Here, when no event is received, the predicted value  $\hat{x}_{RX}(iT_s)$  is within  $\tilde{x}_{TX}(iT_s) \cdot (1 \pm \mu)$ .

As  $(1 + \mu)$  and  $(1 - \mu)$  are always positive,  $x_{rx}(t)$  will always have the same sign as  $x_{rx}(t_{prev})$ . Further generalizing, the value of  $x_{rx}(t)$  after an arbitrary sequence of (eventually mixed in any order) up and down events is  $x_{rx}(t_{after}) = x_{rx}(t_{before}) \cdot (1 + \mu)^N (1 - \mu)^M$ , where N and M are the numbers of up and down events, respectively. As above, given the range of values of  $\mu$ , there is no value of N and/or M (and the range of values of  $\mu$ ).

therefore no sequence of events) which can lead to have the sign of the reconstructed signal different from that of the initial value, i.e., to have the reconstructed signal  $(x_{rx}(t))$  cross the zero. For this reason, we introduce the zero-crossing event.

Algorithm	1:	Custom	event	generation	scheme
Algorium	1.	Custom	CVCIII	generation	SUICH

initialize last event as  $E_{\uparrow}$  $region_sign = 1$ while in acquisition do Compute  $\tilde{x}_{TX}[i]$  by resampling if  $\tilde{x}_{TX}[i]$  is outside the base region then **if** last event was either  $E_{\uparrow}$  or  $E_{\downarrow}$  following  $E_{\uparrow}$ then  $\hat{x}_{TX}[i] = region\_sign \times \theta$ else  $\hat{x}_{TX}[i] \equiv \hat{x}_{\{SoD,Lin,Quad,Avg,PID\}}$ end if sign of  $\tilde{x}_{TX}[i] == sign of \hat{x}_{TX}[i]$  then if  $\mu$  is crossed then if sign of  $x_{TX}^{\Delta}[i] \neq sign$  of  $\tilde{x}_{TX}[i]$  then | Transmit  $E_{\downarrow}$ else Transmit  $E_{\uparrow}$ end else if  $\tilde{x}_{TX}[i-1]$  was inside the base region then Transmit  $E_{\downarrow}$ end end else if  $\tilde{x}_{TX}[i-1]$  was inside the base region then  $region\_sign = region\_sign \times (-1)$ Transmit  $E_{\uparrow}$ end end else if  $\tilde{x}_{TX}[i-1]$  was outside the base region then Transmit  $E_{\uparrow}$ end end end

## C. Zero-crossing

The acceleration samples have signed values, as opposed to the capacitance values of a tactile sensor [13] or the grayscale level of a vision sensor [9], [12]. This characteristic, along with the use of a relative threshold, causes a zero-crossing problem to be addressed. 320

In the communication scheme based on a fixed threshold, the change in the sign of the received value could happen when one out of two possible situations occurs: 323

- 1) an  $E_{\uparrow}$  event is received when  $-\delta(t) < \hat{x}_{RX}(t) < 0$ ; 324
- 2) an  $E_{\downarrow}$  event is received when  $0 \leq \hat{x}_{RX}(t) < \delta(t)$ . 325

By applying the corrections corresponding to the received 326 events and described in equation (6), the sign of the received 327 signal is changed inherently. As a result, only two types 328 of events are required to obtain an unambiguous scheme 329 including the generation of events with an absolute threshold. 330

As opposed to that, a relative threshold-based communica-331 tion scheme does not have this capability, because the correc-332 tion is performed by means of a multiplication by a positive 333 number. This means that the sign of the estimate cannot 334 change, if only  $E_{\uparrow}$  or  $E_{\downarrow}$  polarities are used. In addition, if 335 the estimate reaches zero, the following reconstructed values 336 are going to be zero for the rest of the acquisition. This will 337 be referred to as the "sign change problem" in the following. 338

#### D. Constraining the relative variation of the input 339

An additional problem introduced by the relative threshold 340 is that, when approaching zero, the relative variation of the 341 signal is higher and higher. This reduces the effectiveness of 342 the resampling strategy employed immediately after the sensor, 343 because the signal could double or more from one sample 344 to the next (e.g., increasing from 10 mG to 20 mG). Since 345 doubling is equivalent to an increase of 100%, the expected 346 performance of keeping the relative error under 100µ % cannot 347 be guaranteed ( $\mu$  is usually set below 1). This will be referred 348 to as the "tracking problem" in the following. 349

#### E. Adopted Solution 350

356

357

358

The tracking problem is solved by the introduction of a base 351 threshold  $(\theta)$ , which prevents the system from producing any 352 event whenever the absolute value of the input signal is under 353 that threshold, as shown in Fig. 2. The value of  $\theta$  depends on: 354

- the resolution of the sensor; 355
  - the relative threshold set for the acquisition;
  - where and how the sensor is physically connected;
    - the external environment.

The base threshold could be further tuned after on-field 359 simulations. 360

- On the other hand, the (relative) data threshold  $\mu$  should be 361 a trade-off: 362
- 1) such to obtain a certain maximum relative error out of 363 the base region; 364
- 2) low enough to keep a certain compatibility with respect 365 to the samples produced by the sensor; 366
- 3) high enough to avoid that the event-rate could increase 367 significantly. 368

Once the base region has been introduced, the sign change 369 problem is solved by the use of a third type of event called 370 "Cross Base",  $E_{\uparrow}$ , which may be produced in two different 371 situations: 372

1) the last event produced was an  $E_{\uparrow}/E_{\downarrow}$  event; 373

2) the last event produced was a  $E_{\uparrow}$ . 374

In case 1), the  $E_{\uparrow}$  event communicates to the receiver that 375 the input signal has just gone under the base threshold, causing 376 the estimate to equate  $\pm \theta$ , where the sign is set equal to the 377

TABLE I: Event coding

Event	Outpu
$E_{\uparrow}$	10
$E_{\downarrow}$	01
$E_{\uparrow}$	11
_	00

one of the last predicted value. In case 2), a second  $E_{\uparrow}$  after 378 another communicates both the exit from the base region and 379 the change in the sign of the function. Until the next event, 380 the estimate will be  $\pm \theta$ . 381

If, in case 1), after  $E_{\uparrow}$  the signal exits the base region with 382 the same sign, a  $E_{\downarrow}$  is produced, to allow the algorithm to 383 restart computing the estimate with the implemented estima-384 tion algorithm. This solves the sign change problem without 385 introducing ambiguities in the communication scheme.

Algorithm 1 reports the methodology used to implement the 387 complete communication scheme, including the conditions for 388 the  $E_{\uparrow}$  generation. Because there are three possible events and 389 the no event situation, two output lines are used to code the 390 events, as reported in Table I. 39

#### III. TAXONOMY OF ASYNCHRONOUS ALGORITHMS

Several asynchronous sampling algorithms have been inves-393 tigated [1]-[4]. In the application under investigation, the in-394 troduction of the dynamic threshold and the choice of sending 395 just events and not the whole sample reduce the quantity of 396 the potentially working methods to the magnitude-driven ones 397 only. At the beginning, even the integral algorithms have been 398 considered. Within this category Send on Area [6] and Send 399 on Energy [7] are well-known methods. However, they are 400 based on an integral relationship with respect to the original 401 signal, so the threshold used concerns the primitive function 402 in the Send on Area and the primitive of the signal squared in 403 the Send on Energy. This prevents the conceived scheme from 404 limiting the maximum relative error on the reconstruction of 405 the signal itself, thus this category of algorithms has not been 406 considered in the following. 407

On the other hand, the algorithms of the magnitude-driven 408 category are characterized by a direct relationship between the 409 approximating error and the received signal itself. This feature 410 is necessary because all the methods available in the literature 411 are based on a static absolute threshold and on the transmission 412 of the whole data, whereas the main requirement for the 413 system under development is to reduce the output bandwidth. 414 Indeed, with the event-based communication scheme defined 415 in Section II, only 2 bits per spatial axis would be necessary. 416

Further limitations in the choice of suitable sampling al-417 gorithms depend on the need to limit the complexity of the 418 hardware implementation and optimization of resources. After 419 taking into account both the mathematical and complexity 420 criteria, the following algorithms have been analyzed: 421

A. Send on Delta;	422
B. Linear;	423
C. Quadratic;	424

D. Average; 425

386



Fig. 2: Zero crossing and base threshold  $\theta$ : when the signal is close to zero, the relative thresholding can cause the generation of way too many events, as the smaller the initial signal is, the finest the sensitivity of the thresholding is. This is solved by setting a "base" threshold. When the signal falls within  $+/-\theta$ , no events are generated. (a,b) clarify the two cases described in the main text, when the signal changes sign.

E. Proportional-Integral-Derivative (PID). 426

#### A. Send on Delta 427

The Send on Delta (SoD) is the most popular algorithm and 428 uses the last received value corresponding to the last received 429 event, as the estimatefor the following samples, until another 430 event is received. 431

The approximation  $\hat{x}_{SoD}[i]$  yielded by this algorithm cor-432 responds to a zero-order estimation and the formula is inde-433 pendent of the time elapsed from the last event [1]: 434

$$\hat{x}_{SoD}[i] \triangleq \hat{x}[i] = \hat{x}[i_L] \tag{11}$$

with  $i_L$  the index of the last event. 435

#### B. Linear 436

The Linear method is based on a first-order approximation 437  $\hat{x}_{Lin}[i]$  of the signal [2]: 438

$$\hat{x}_{Lin}[i] \triangleq \hat{x}[i] = \hat{x}[i_L] + \frac{\hat{x}[i_L] - \hat{x}[i_{L-1}]}{i_L - i_{L-1}} \cdot (i - i_L) \quad (12)$$

with  $i_{L-1}$  the index of the last but one event. 439

Differently from the SoD algorithm, the time elapsed from 440 the last and last but one events is used to compute the estimate, 441 so two events are required to compute the estimate. When no 442 event is available or the signal is inside the base region, the 443 estimation is based on the Send-On-Delta algorithm with  $\pm \theta$ 444 as an estimate. When only one event is available (e.g., after 445 exiting the base region or after the very first received event), 446 the method uses it as  $x^*(t_L)$  and  $\pm \theta$  as  $x^*(t_{L-1})$ . 447

#### C. Quadratic 448

The Quadratic algorithm, characterized by  $\hat{x}_{Quad}[i]$ , in-449 creases the approximation by one order with respect to the 450 Linear method [2]: 451

$$\hat{x}_{Quad}[i] \triangleq \hat{x}[i] = \hat{x}[i_L] + \frac{\hat{x}[i_L] - \hat{x}[i_{L-1}]}{i_L - i_{L-1}} \cdot (i - i_L) + \\
+ \frac{1}{2} \left[ \frac{\hat{x}[i_L] - \hat{x}[i_{L-1}]}{(i_L - i_{L-1})^2} - \frac{\hat{x}[i_{L-1}] - \hat{x}[i_{L-2}]}{(i_L - i_{L-1}) \cdot (i_{L-1} - i_{L-2})} \right] \cdot \\
\cdot (i - i_L)^2$$
(13)

Like the Linear algorithm, this method has an intrinsic 452 latency, as at least three events out of the base region should have been produced to apply the algorithm.

In addition, not only does this method require to track the 455 time elapsed from the last  $(i_L)$  and last but one events  $(i_{L-1})$ , 456 it also needs the time elapsed from the last but two event 457  $(i_{L-2})$ . Until two events are available, it behaves like the 458 Linear algorithm; when two events are available,  $\hat{x}[i_{L-2}]$  is 459 set to  $\pm \theta$ . 460

#### D. Average

The Average method obtains the estimate  $\hat{x}_{Avg}[i]$  with a 462 summation over the last M predicted values, divided by M. 463 When all the last M predicted values are equal to each other, 464 this method is equivalent to the SoD [3]: 465

$$\hat{x}_{Avg}[i] \triangleq \hat{x}[i] = \frac{1}{M} \sum_{j=0}^{M-1} \hat{x}[i-j]$$
 (14)

M is chosen depending on the desired number of predicted 466 values one would like to consider and the number of available 467 memory elements, to store the previous predicted values. In 468 the simulations, M has been set to three, as suggested in [3]. 469 If less than M values are available, the average is computed 470 on the available number. 471

The PID algorithm is based on the same theory as in the 473 Control Application field. It groups the Send on Delta, Average 474 and Linear methods (this last one in a further approximate 475 form, to avoid considering the elapsed times), weighing their 476 contributions differently [3] into  $\hat{x}_{PID}[i]$ : 477

453 454

461



Fig. 3: Event-driven encoding of a signal over time. Top: acquired signal (subset of samples from dataset #8) and reconstructed waveforms using each algorithm, the black lines represent the boundaries of the base region. Bottom: Events produced by each algorithm with the transmitter fed with the acquired signal. No event is produced during the time interval when the acquired signal is in the base region. At time 0.17 s all the algorithms produce an  $E_{\pm}$  when the acquired signal is sampled with opposite sign with respect to the estimate just outside the base region. Consequently, the reconstructed signal changes from  $+\theta$  to  $-\theta$  (falling edge at 0.18 s).

$$\hat{x}_{PID}[i] \triangleq \hat{x}[i] = w_{SoD} \cdot \hat{x}[i_L] + w_{Aver} \cdot \frac{1}{M} \sum_{j=0}^{M-1} \hat{x}[i-j] + w_{Lin} \cdot (\hat{x}[i_L] - \hat{x}[i_{L-1}])$$
(15)

As suggested in [3], the three coefficients have been set to: 478

- 1)  $w_{SoD} = 0.4;$ 479
- 2)  $w_{Aver} = 0.6;$ 480
- 3)  $w_{Lin} = 0.3$ . 481

485

When less than M events are available, the estimate s com-482 puted for each of the three contributions as previously detailed 483 in the descriptions of the corresponding algorithms. 484

#### IV. METHODOLOGY

The algorithms have been written in Matlab<sup>®</sup> code and 486 they were simulated with twenty-one heterogeneous datasets 487 as input stimuli (Table II). The first three datasets listed were 488 used because already present in the technical literature [2], 489 [3] and to show that the implemented scheme is general and 490 could be employed on control and medical waveforms as well. 491 The datasets related to the sensor were obtained from real-492 time acquisitions using a general-purpose computer as data 493 collector from the FPGA, which, in turn, receives data from 494 the accelerometer. Three different situations were used for 495

TABLE II: Datasets Used for the Algorithm Test.

Dataset #	Content	
1	1 <sup>st</sup> order response	
2	$2^{nd}$ order response	
3	Electro-cardiography	
4-6	Sensor manual static XYZ	
7-9	Sensor manual tilting XYZ	
10-12	Sensor manual shaking XYZ	
13-15	Robot motion XYZ	
16-18	Robot static XYZ	
19-21	Robot shaking XYZ	

generating the accelerations: a static one, one tilting the sensor 496 along its Z axis and one shaking the sensor along its X axis. 497 The robotic datasets were acquired on the iCub robot left hand in three different movement conditions, similarly to the sensor data: one static, one moving the arm randomly and one shaking the arm.

The events produced have been collected by a software 502 receiver, which reconstructs the original waveform within 503 a maximum relative error equal to the maximum relative 504 threshold set. The reconstructed waveforms, along with the 505 polarity of the received events for a portion of dataset #8, are 506 shown in Figure 3. 507



Fig. 4: Results of the simulations: a)  $\gamma_{abs}$ , b)  $\gamma_{rel}$ , c)  $\xi_{abs}$  and d)  $\sigma$ .

The performance was evaluated in terms of accuracy, mea-508 sured as error on the reconstructed waveform (i.e., the error 509 between the  $\tilde{x}_{RX}(t)$  computed by the RX and the signal 510  $\tilde{x}_{TX}(t)$  internally used by the TX to generate the events), 511 and effectiveness, measured as the reduction in the number of 512 output messages in the conceived event-driven scheme with 513 respect to a synchronous transmission. For the sake of clarity 514 and without loss of generality, we assume no latency and 515 perfect synchronization (same timings) between TX and RX, 516 so that we can use the two corresponding discrete-time signals 517  $\tilde{x}_{TX}[i]$  and  $\tilde{x}_{RX}[i]$ . In all the simulations,  $\mu$  is set to 0.02, in 518 order to have a maximum relative error of 2 %. This value has 519 been chosen as an acceptable tradeoff between reduced data 520 rate and resulting accuracy. 521

<sup>522</sup> Moreover, following the points listed in Section II-C and <sup>523</sup> the specifications of the employed accelerometer [23],  $\theta$  is set <sup>524</sup> to 0.12 G.

## 525 A. Accuracy figures

529

The first two performance figures, concerning the accuracy evaluation, have an average meaning: the average absolute error ( $\gamma_{abs}$ ) and the average relative error ( $\gamma_{rel}$ ) [2].

$$\gamma_{abs} = \frac{1}{N_{samples}} \cdot \sum_{i=n_{in}}^{n_{fin}} |\tilde{x}_{TX}[i] - \tilde{x}_{RX}[i]| \qquad (16)$$

$$\gamma_{rel} = \frac{1}{N_{samples}} \cdot \sum_{i=n_{in}}^{n_{fin}} \frac{|\tilde{x}_{TX}[i] - \tilde{x}_{RX}[i]|}{|\tilde{x}_{TX}[i]|}$$
(17)

where  $n_{in}$  and  $n_{fin}$  are the first sample index and the last one, respectively. The average relative error is a factor with respect to the reference. Moreover, the maximum absolute  $_{532}$   $(\xi_{abs})$  and relative  $(\xi_{rel})$  errors have been evaluated.  $_{533}$ 

$$\xi_{abs} = max \left( \left| \tilde{x}_{TX}[i] - \tilde{x}_{RX}[i] \right| \right)$$
(18)

$$\xi_{rel} = max \left( \frac{|\tilde{x}_{TX}[i] - \tilde{x}_{RX}[i]|}{|\tilde{x}_{TX}[i]|} \right) \tag{19}$$

As the signal changes sign, another useful parameter is the root mean squared error ( $\sigma$ ): 536

$$\sigma = \sqrt{\frac{\sum_{i=n_{in}}^{n_{fin}} \left| \tilde{x}_{TX}[i] - \tilde{x}_{RX}[i] \right|^2}{N_{samples}}}$$
(20)

#### B. Effectiveness figures

537

Two performance figures evaluate the gain of the algorithm, where the gain is the number of samples saved by using the event-based method instead of a synchronous one. 540

The first figure is the equivalent sampling rate (called m 541 in [6]), which is the reciprocal of the average of the interevent intervals: 543

$$m = \frac{1}{\Delta t_{aver}} = \frac{N_{events}}{\sum_{i=n_{in}}^{n_{fin}} \Delta t_i}$$
(21)

The second figure is called Effectiveness (also called *energy* <sup>544</sup> *ratio* as in [3]): <sup>545</sup>

$$E = \frac{N_{samples}}{N_{events}} \tag{22}$$

For example, an Effectiveness equal to  $\sim 10^3$  means that 1 second synchronous samples is sent. 547

553

#### C. Hardware considerations 548

The maximum available latency is 50 clock cycles because 549 the input frequency equals 5 MHz from the resampler and the 550 internal working frequency of the target system is 250 MHz 551 (both frequencies are compliant with [20]). 552

V. SIMULATION RESULTS

As detailed in the previous paragraphs, the proposed com-554 munication system is made of different blocks, each of which 555 has been sized as follows. The resampler unit has 16 fractional 556 bits of precision; the integer part depends on the sensor full 557 scale, which had been set to the maximum available on the 558 specific accelerometer employed in this work, i.e.  $\pm 16 \text{ G}$  [23]. 559 As a consequence, 6 bits of integer part are required in a 560 signed representation to properly encode even the +16 G value. 561 Moreover, the Linear and Quadratic algorithms also need the 562 inter-event time to be used for the computation, see Eqs. 12) 563 and 13). A 1s inter-event time could be considered high 564 enough to avoid any timer wrapping, when the acceleration 565 is above the base threshold. In case the acceleration is within 566 the base threshold, a time wrap does not cause any problem 567 because the estimation considers only the base threshold. For 568 a time counter updated at every new sample coming from the 569 resampler, 1 s corresponds to 5 million ticks, i.e., 24 signed 570 bits. For this reason, the integer part of the data is extended 571 to 8 bit and the acceleration data parallelism is set to [8.16] 572 fixed-point representation. As a consequence, the maximum 573 time interval without wrapping is  $2^{23} - 1$ , about 1.68 s. 574

Figures 4 and 5 show the performance of the different EG 575 algorithms over all the considered datasets. For the accuracy 576 figures ( $\gamma_{abs}$ ,  $\gamma_{rel}$ ,  $\xi_{abs}$  and  $\sigma$ ), the lowest box represents the 577 one which, statistically on the considered datasets, has the best 578 behavior. Conversely, for the effectiveness figures, the situation 579 changes between the equivalent sampling rate (m) and the 580 Effectiveness (E). The lower the equivalent sampling rate, the 581 better, whereas the lower the Effectiveness the worse. As a 582 preliminary evaluation, the maximum relative error has been 583 verified to be under the desired threshold for all the algorithms. 584

#### A. Accuracy parameters analysis 585

In terms of Average Absolute Error, the best algorithms are 586 the Linear and Quadratic ones, with the latter being slightly 587 better than the former. Their values extend from 0.01 G to 588 0.085 G, approximately. The SoD comes immediately after-589 wards, whereas the Average and PID have a wider dispersion 590 towards higher values, greater than 0.1 G, with the PID being 591 slightly better than the Average. 592

In terms of Average Relative Error, the best algorithms are, 593 again, the Linear and Quadratic ones, but in this case the 594 former extends more than the latter in the low-value zone of 595 the error, arriving at 0.4%, with the upper boundary at 0.7%. 596 The Send on Delta is again the third best algorithm, exhibiting 597 a very narrow dispersion around 0.9%. The Average remains 598 the worst, followed by the PID, extending over 1.2%. 599

For what concerns the Maximum Absolute Error, the dis-600 tributions are almost the same for all the algorithms, with the 601 25<sup>th</sup> and 75<sup>th</sup> percentiles between 0.06 G and 0.37 G. 602

The Root Mean Squared Error shows a situation very similar 603 to the Average Absolute Error case, where the Linear and 604 Quadratic algorithms are very close to each other, with a 605 distribution between 0.02 G and 0.12 G. The Average is still 606 the worst one, followed by the PID and the SoD. 607

#### B. Effectiveness parameters analysis

For the effectiveness figures, the Equivalent Sampling Rate 609 shows a very narrow distribution for the Linear and Quadratic 610 algorithms, meaning that their gain is almost independent of 611 the trend of the input signal. The Linear is slightly better be-612 cause its box extends partially under the one of the Quadratic. 613 The SoD extends more towards high values, until 576 events/s, 614 approximately. The Average remains the worst one even in 615 this case, followed by the PID. 616

In the end, for the Effectiveness, the first and second order 617 waveforms show that the SoD, Linear and Quadratic have the 618 same performances. However, when considering the remaining 619 10 datasets a significant difference is shown between the 620 Linear-Quadratic pair and the SoD algorithms: if the for-621 mer extends between 2.9.104 samples/event and 7.0.104 samples/event, 622 the latter box is comprised between 2.1.10<sup>4</sup> samples/event and 623 4.8.10<sup>4</sup> samples/event. The SoD is below, and the Average and 624 PID methods are even lower than the other ones. 625

More in detail, the performance of the SoD equals the ones of the Linear and Quadratic methods for the datasets acquired with a manual movement of the accelerometer. On the other hand, the Linear and Quadratic algorithms are always better than the SoD on the robotic datasets, with:

- Accuracy parameters: -0.4 % up to -26.7 % over SoD (-5.4% avg.);
- Equivalent sampling rate: -7 % up to -58 % over SoD (-20.0 % avg.);
- Effectiveness: +8 % up to +142 % over SoD (+48.0% avg.).

This analysis allows to conclude that the best methods 637 are the Linear and Quadratic, followed by the SoD. As the other two algorithms are more complex and show worse performance than the SoD, they are not addressed in the 640 following part of this work.

## VI. HARDWARE IMPLEMENTATION

The conceived algorithmic scheme has been described in 643 VHDL to be implemented onto an FPGA platform. In order 644 to achieve the target operating frequency, pipeline stages have 645 been added inside the hardware block. The number of pipeline 646 stages from input to output is however limited from the 50 647 clock cycles latency requirement. The available FPGA is a 648 speed-grade 1, the lowest speed-grade for the Xilinx Artix-7 649 XC7A35T model. As a consequence, the number of pipeline 650 stages was tuned until the target frequency was met and the 651 latency requirement was not exceeded (see Section IV-C). This 652 led to a  $\frac{N}{2}$  pipe stages inside the multipliers and N inside 653 the divider employed in the Linear and Quadratic algorithms, 654 with N = 24 as detailed in Section V. While for the Linear the 655 total latency in the worst case (the one requiring the estimation 656 correction) is within 50 clock cycles, the Quadratic exceeds 657

608

629 630 631

632

633

634

635

636

638

639

641

642

626

627



Fig. 5: Results of the Matlab<sup>®</sup> simulations: a) Equivalent Sampling Rate (m), b) Effectiveness related to the first two datasets and c) Effectiveness for all the remainder datasets.

this limit. Moreover, the performance parameters show that the 2<sup>nd</sup> order term does not give any appreciable advantage over the Linear algorithm. These two considerations allow to discard the Quadratic algorithm.

Finally, considering both the algorithmic performance and hardware optimization, the Linear and Send-on-Delta algorithms appear the most suitable to be mapped onto the FPGA.

Since the employed transducer is a tri-axis accelerometer, 665 the hardware blocks implementing the resampling and the 666 estimation have been replicated once per each axis. The 667 complete architecture is shown as a block diagram repre-668 sentation in Figure 6. The SPI Manager block acquires the 669 synchronous samples from the accelerometer by using a 4-670 wire SPI communication. Then, it splits the data of each 671 axis and feeds them to the corresponding resampling block. 672 The output event lines consist, each, in a 6-bit address that 673 identifies the axis and the accelerometer, followed by the 2-674 bit event information in the LSBs. Keeping the address inside 675 the transmitted data is useful in case the event stream from the 676 accelerometer shares the output channel of the system, where 677 it is embedded, with other event streams coming from different 678 sensors. This allows the receiving end to acknowledge the 679 source of the arriving events and is based on the Address-680 Event-Representation (AER) [15]–[17]. Moreover, if the 3 681 event lines are multiplexed at the transmitter, only 8 bits 682 instead of 24 are present, further reducing the routing cost. 683

Figure 7 shows the hardware arrangement. The MEMS accelerometer is connected to an Arty board, hosting an Artix-7 FPGA. The connection is achieved by using a series of jumpers toward one of the PMOD connectors of the Arty board. The board is also connected to a general-purpose computer with two cables: the black one in Figure 7 is a USB cable, used to program the FPGA and to receive the raw acceleration samples from the FPGA; the white one is an Ethernet cable, for collecting the events from the FPGA. The Ethernet payload contains the data coded in AER.

A. Results

Table III shows the post-implementation complexity (esti-695 mated by Xilinx Vivado<sup>®</sup> software) for the single EG block 696 when it implements either the SoD or the Linear algorithm. 697 By replicating it 3 times and adding the SPI Manager block 698 and the resamplers, the LUT usage is 2892 elements (13.9% 699 of the total) and the FF usage is 3620 elements (8.7% of 700 the total) for the Linear case. The block works reliably at the 701 target frequency of 250 MHz. 702

In order to evaluate the improvement with respect to a 703 synchronous system, the output bandwidth is estimated as: 704

$$BW_o = o_b \sum(m) \tag{23}$$

where  $o_b$  is the number of output bits and  $\sum (m)$  is the sum of the equivalent sampling rates obtained on each axis, due to the multiplexed output. It is substituted with the fixed output data rate of the accelerometer in the synchronous system.

## B. Hardware setup

The synchronous system has a multiplexed output, for a symmetrical comparison to the event-driven one. As a result, 711 the synchronous system has an 18-bit output data (6-bit address and 12-bit acceleration data). The worst-case situation 713 for the Manual acquisitions is the shaking case (datasets 10-714 12) with  $\sum (m) = 753.9 \text{ events/s}$ . The worst-case situation for 715 (m) = 753.9 events/s.

709



Fig. 6: Block diagram of the complete hardware architecture and relative resource usage of every block after implementation on a Xilinx XC7A35T-L1 FPGA.



Fig. 7: Hardware setup for the real-time acquisition.

the Robotic datasets is the static case (datasets 13-15) with  $\sum (m) = 1698.0$  events/s. The synchronous data, corresponding to the previous two categories of datasets, are produced by the accelerometer at 1344 Hz.

Figure 8 shows the reconstructed waveforms using SoD and Linear algorithms on ten out of the twenty-one considered datasets.

The performance of the algorithmic block on a single-output data are also considered for the first 3 datasets, which are generic waveforms. In that case the worst-case equivalent sampling rate is 229.5 events/s, obtained with the Medical waveform (dataset #3, Figure 8c). The synchronous data rate for that dataset is 360 Hz. The  $BW_o$  variation is shown in Table IV, when switching from a synchronous transmission to an event-

TABLE III: Complexity of the implemented algorithms on Xilinx XC7A35T-L1 FPGA.

Block/Port	SoD	Linear
LUTs	582 (2.8 %)	842 (4.0 %)
FF	624 (1.5 %)	969 (2.3 %)
BRAM	0 (0 %)	0 (0 %)
IO	30 (14.3 %)	30 (14.3 %)
BUFG	2 (6.3 %)	2 (6.3 %)
MMCM	1 (20 %)	1 (20 %)

driven one operating on the same input datasets. 730 As it can be observed, when the movement has a full 731



Fig. 8: Example of the considered datasets. the waveforms include the input resampled data (blue), the reconstructed signal with a Linear algorithm (green), the reconstructed signal with a SoD algorithm (red) : a) dataset #1, b) dataset #2, c) dataset #3, d) dataset #4, e) dataset #7, f) dataset #10, g) dataset #14, h) dataset #16, i) dataset #19, j) dataset #21.

TABLE IV: Output Bandwidth Comparison.

Datasets	Synchr. (kbps)	Event-dr.(kbps)	Difference
Manual	24.2	6.0	-75 %
Robotic	24.2	13.6	-44 %
CNTRL & Medical	6.5	1.8	-72 %

dynamic within  $\pm 2$  G, i.e., for a manual movement, the SoD yields the same performance as the Linear, being very attractive for its reduced complexity. In the robotic equipment, instead, the Linear algorithm performs better as discussed in Section V.

## VII. CONCLUSION

737

This paper has detailed the entire process aimed at conceiving a new event-driven scheme, identifying the best eventgeneration algorithm from accuracy, effectiveness and implementation points of view. Both the communication scheme and the algorithm were mapped onto an FPGA hardware to code the information received from a digital MEMS accelerometer. Experimental results show that the best estimation algorithm is the first-order, or Linear, one. The real-time acquisitions 745 show that the maximum relative error is kept bounded within 746 the desired relative threshold set at the beginning of the 747 acquisition. The total resource usage in a low-cost FPGA does 748 not exceed the 15% of both the logic cells and flip-flops. The 749 output bandwidth, thanks to both the 8-bit output event coding 750 and the obtained Effectiveness, is reduced by more than 40%751 in all the acceleration datasets, with a -44 % improvement in 752 the robotic platform case. An improvement of -72% is also 753 observed for single waveforms with slower synchronous data 754 rate. 755

Although the analysis of the event-driven generation meth-756 ods has been performed using an accelerometer as input 757 device, the formulation, characterization and FPGA imple-758 mentation are general enough to hold for different types of 759 sensors, as needed in a fully event-driven robotic sensing 760 system. The main advantages of the event-driven approach 761 are low latency and compression. Our accuracy results show 762 that the compression does not decrease the information content 763 gathered by the sensor. The challenge in the design of artificial 764 perception based on this principle is that of finding principled 765 methods to extract this information without resorting to signal 766

reconstruction and "traditional" algorithms. A fair amount of 767 work has been published so far on the development of such 768 methods, mostly on vision for robots. All the methods and 769 results reported so far show that not only "frame" recon-770 struction is unnecessary, but also that those algorithms are 771 robust to noise and to the loss of few events (hence having 772 some drop in accuracy). This work goes in the direction of 773 making other sensory modalities available to roboticists to de-774 velop multi-modal perception systems that improve efficiency, 775 robustness and autonomy of robots, developing event-driven 776 multi-sensory perception algorithms, ready for when native 777 event-driven sensors will be mature enough to be integrated in 778 robots. In the specific case of accelerometers, the information 779 gathered from the sensor will be useful to assess the movement 780 of the robot, or to detect impact on surfaces, or to classify 781 roughness of surfaces from vibrations. In the first case, the 782 latency of the signal is crucial to detect the contact and correct 783 the action; in the second case, the frequency content of the 784 signal, rather than the acceleration instantaneous value, is 785 important. To study all these aspects, future work includes 786 the integration of the designed system into the iCub robot 787 and the substitution of the Ethernet output with the custom 788 serial protocol discussed in [20], for event transmission in 789 AER packets. 790

Additionally, the design is fully-portable to other platforms, 791 like ASICs. We will hence develop custom chips with event-792 driven event generation for off-the-shelf sensors to reduce size 793 and power consumption due to the use of the FPGA. Further 794 developments include the implementation of neural algorithms 795 instead of asynchronous sampling ones and the design of an 796 event-driven readout accelerometer which uses the conceived 797 communication scheme and algorithm internally. 798

799

802

#### AUTHORS' NOTICE

The material (i.e., the source code and datasets) presented in this work could be provided by the authors upon request.

## REFERENCES

- [1] M. Miskowicz, "Send-on-delta concept: An event-based data reporting
   strategy", in *Sensors*, v. 6, pp. 49-63, 2006.
- [2] K. Staszek, S. Koryciak, M. Miskowicz, "Performance of send-on-delta
   sampling schemes with prediction", in *Proceedings of IEEE Interna- tional Symposium on Industrial Electronics ISIE 2011*, pp. 2037-2042,
   June 2011.
- [3] F. Xia, Z. Xu, L. Yao, W. Sun, M. Li, "Prediction-based data transmission for energy conservation in wireless body sensors", in *Proceedings of Wireless Internet Conf. WICON 010*, pp. 1-9, March 2010.
- [4] Various authors, *Event-Based Control and Signal Processing*, ed. By M.
   Miskowicz, CRC Press, 2016.
- [5] M. Simonov, G. Chicco and G. Zanetto, "Event-Driven Energy Metering: Principles and Applications", in *EEE Transactions on Indus- try Applications*, vol. 53, no. 4, pp. 248–251, July-Aug. 2017. doi: 10.1109/TIA.2017.2679680
- [6] M. Miskowicz, "The event-triggered integral criterion for sensor sampling", in *Proceedings of IEEE International Symposium on Industrial Electronics ISIE 2005*, v. 3, pp. 1061-1066, November 2005.
- [7] M. Miskowicz, "Efficiency of event-based sampling according to error energy criterion", in *Sensors*, v. 10, n. 3, pp. 2242-2261, 18 Mar. 2010.
- [8] Posch, C., Matolin, D. and Wohlgenannt, R., "A QVGA 143dB Dynamic Range Asynchronous Address-Event PWM Dynamic Image Sensor with Lossless Pixel-Level Video Compression", in 2010 IEEE International Conference on Solid-State Circuits (ISSCC), 7-11 Feb. 2010.

- [9] Posch, C., Matolin, D. and Wohlgenannt, R., "An asynchronous timebased image sensor", in 2008 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2130-2133, 18-21 May 2008.
- [10] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128×128 120dB 30mW asynchronous vision sensor that responds to relative intensity change", in 2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers, San Francisco, CA, 2006, pp. 2060-2069.
- [11] V. Chan, S.-C. Liu and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface", in *IEEE Transactions on Circuits and Systems I*, 2007, pp. 48-59, vol. 54(1).
- [12] B. Son, Y. Suh, S. Kim, H. Jung, J. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsiannikov and H. Ryu, "4.1 A 640×480 dynamic vision sensor with a 9um pixel and 300Meps address-event representation", in 2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, 2017, pp. 66-67.
- [13] E. Baglini, G. Cannata and F. Mastrogiovanni, "Design of an embedded networking infrastructure for whole-body tactile sensing in humanoid robots" in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010.
- [14] L. Osborn, R. R. Kaliki, A. B. Soares and N. V. Thakor, "Neuromimetic event-based detection for closed-loop tactile feedback control of upper limb prostheses", in *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 196–206, April 2016.
- [15] M. Sivilotti "Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks", 1991.
- [16] A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels and B. Linares-Barranco, "On synthetic AER generation", in 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512), 2004, pp. V-784-V-787 Vol.5.
- [17] R. Paz-Vicente, A. Linares-Barranco, A. Jimenez-Fernandez, G. Jimenez-Moreno and A. Civit-Balcells, "Synthetic retina for AER systems development", in 2009 IEEE/ACS International Conference on Computer Systems and Applications, Rabat, 2009, pp. 907-912.
- [18] P. Motto Ros, M. Crepaldi, and D. Demarchi, "A hybrid quasidigital/neuromorphic architecture for tactile sensing in humanoid robots", in Advances in Sensors and Interfaces (IWASI), 2015 6th IEEE International Workshop on Advances in Sensors and Interfaces, June 2015.
- [19] P. Motto Ros, M. Crepaldi, C. Bartolozzi and D. Demarchi, "Asynchronous DC-free serial protocol for event-based AER systems", in 2015 *IEEE International Conference on Electronics, Circuits, and Systems* (*ICECS*), Cairo, 2015, pp. 248–251. doi: 10.1109/ICECS.2015.7440295
- [20] C. Bartolozzi, P. Motto Ros, F. Diotalevi, N. Jamali, L. Natale, M. Crepaldi and D. Demarchi, "Event-driven encoding of off-the-shelf tactile sensors for compression and latency optimisation for robotic skin", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 166-173.
- [21] S. Caviglia, L. Pinna, M. Valle, and C. Bartolozzi, "An event-driven POSFET taxel for sustained and transient sensing", in 2016 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 349-352, May 2016.
- [22] F. Bergner, E. Dean-Leon, and G. Cheng, "Event-based signaling for large-scale artificial robotic skin — realization and performance evaluation", in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016), October 2016.
- [23] ST Microelectronics, *LIS3DH MEMS digital output motion sensor: ultra-low-power high-performance 3-axis "nano"accelerometer*, December 2016.
- [24] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. Victor, F. Becchi, G. Sandini, G. Metta, "The design of the iCub humanoid robot", in *International Journal of Humanoid Robotics*, vol. 09, no. 04, 1250027, 2012.

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

Marino Laterza is Fellow researcher at Istituto Italiano di Tecnologia.
 He received the Bachelor's Degree in Electronic Engineering (Summa cum

- Laude) in 2015 and the Master of Science in Electronic Engineering in 2017
- 891 (Summa cum Laude), both from Politecnico di Torino, Italy. His interests
- regard digital hardware design on both programmable and application-specific platforms. His current research is focused on the implementation of neural
- <sup>894</sup> models on FPGA to emulate human perception in tactile sensing.

Paolo Motto Ros Paolo Motto Ros is Senior Post-Doc researcher at Istituto 895 Italiano di Tecnologia, Electronic Design Laboratory. He received the elec-896 tronic engineering degree and the Ph.D. in electronic engineering from the 897 Politecnico di Torino (Polito), Turin, Italy, in 2005 and 2009, respectively. 898 From 2009 to 2012 he was with Neuronica Laboratory (Dipartimento di Elet-899 tronica, Politecnico di Torino) as Post-Doc, working on assistive technologies, 900 computer vision and learning machines projects. He joined the Center for 901 Space Human Robotics (CSHR), Istituto Italiano di Tecnologia (IIT), Turin, 902 Italy, in 2012; since 2016 he is with Electronic Design Laboratory (EDL), 903 IIT, Genoa, Italy. He is an IEEE member since 2016, and member of the 904 Circuits And Systems (CAS) society. He counts >20 publications; current 905 research include design of full-custom ultra-low-power asynchronous digital 906 907 integrated circuits, event-based smart-sensors, sensor networks, bio-inspired 908 electronics, neuromorphic engineering.

Danilo Demarchi Associate Professor at Politecnico di Torino, Italy, De-909 partment of Electronics and Telecommunications. Lecturer at EPFL Lau-910 sanne, Switzerland, Adjunct Professor at Tongji University Shanghai, China, 911 Associate Faculty at the University of Illinois at Chicago, Department of 912 Electrical and Computer Engineering. Leading the MiNES (Micro&Nano 913 Electronic Systems) Laboratory of Politecnico di Torino. Senior Member of 914 IEEE, Member of the BioCAS Technical Committee, Associate Editor of 915 the Transactions on Biomedical Circuits and Systems (TBioCAS), Associate 916

Editor of IEEE Sensors and of the Springer Journal BioNanoScience. General
 Chair of BioCAS 2017 (Biomedical Circuits and Systems) Conference edition

919 in Torino.

Maurizio Martina Maurizio Martina (IEEE Senior Member) is associate
 professor at Politecnico di Torino. He received the electronic engineering
 degree and the Ph.D. in electronic engineering from the Politecnico di
 Torino in 2000 and 2004, respectively. From 2004 he is with the VLSI lab (Electronics and Telecommunications Department, Politecnico di Torino)
 working on VLSI architectures for digital signal processing. He is author and
 coauthor of about 100 scientific papers.

Chiara Bartolozzi Chiara Bartolozzi (IEEE Member) is researcher at the 927 Istituto Italiano di Tecnologia. She earned a degree in Engineering (with 928 honors) at University of Genova (Italy) and a Ph.D. in Neuroinformatics 929 at ETH Zurich, developing analog subthreshold circuits for emulating bio-930 physical neuronal properties onto silicon and modelling selective attention on 931 932 hierarchical multi-chip systems. She is currently leading the Event Driven Perception for Robotics group (www.edpr.iit.it), mainly working on the appli-933 cation of the "neuromorphic" engineering approach to the design of sensors 934 935 and algorithms for robotic perception. She is chair of the Neuromorphic Systems and Application Technical Committee of IEEE CAS. 936