Development of an acoustically adaptive modular system for near real-time clarity-enhancement

Liu Cheng, Alexander; Cruz, Patricio; Llorca Vega, Nestor; Mena, Andrés

DOI
10.1007/978-3-030-34255-5_12

Publication date
2019

Document Version
Final published version

Published in
Ambient Intelligence

Development of an Acoustically Adaptive Modular System for Near Real-Time Clarity-Enhancement

Alexander Liu Cheng[1,2(✉)], Patricio Cruz[3], Nestor Llorca Vega[2,4], and Andrés Mena[2]

[1] Faculty of Architecture, Delft University of Technology, Delft, The Netherlands
a.liucheng@tudelft.nl
[2] Faculty of Architecture and Engineerings, Universidad Internacional SEK, Quito, Ecuador
[3] Faculty of Electrical and Electronic Engineering, Escuela Politécnica Nacional, Quito, Ecuador
[4] School of Architecture, Universidad de Alcalá, Madrid, Spain

Abstract. This paper details the development of an acoustically adaptive modular system capable of enhancing Speech Clarity (C_{50} Clarity Index) in specific locations within a space in near real-time. The mechanical component of the system consists of quadrilateral, truncated pyramidal modules that extend or retract perpendicularly to their base. This enables said modules (1) to change in the steepness of the sides of their frustum, which changes the way incoming sound waves are deflected/reflected/diffused by the surfaces of the pyramid; and (2) to reveal or to hide the absorbent material under each module, which enables a portion of incoming sound waves to be absorbed/dissipated in a controlled manner. The present setup considers a fragmentary implementation of six modules. The behavior of these modules is determined by two steps in the computational component of the system. First, the initial position of the modules is set via a model previously generated by an evolutionary solver, which identifies the optimal extension/retraction extent of each of the six modules to select for individual configurations that collectively ascertain the highest clarity in said specific locations. Second, a simulated receiver at the location in question measures the actual clarity attained and updates the model's database with respect to the configuration's corresponding clarity-value. Since the nature of acoustics is not exact, if the attained measurement is lower than the model's prediction for said location under the best module-configuration, but higher than the second-best configuration for the same location, the modules remain at the initial configuration. However, if the attained values are lower, this step reconfigures the modules to instantiate the second—or third-, fourth-, etc.—best configuration and updates the model's database with respect to the new optimal module-configuration value. These steps repeat each time the user moves to another specific location. The objective of the system is to contribute to the intelligent and intuitive Speech Clarity regulation of an inhabited space. This contributes to its *Interior Environmental Quality*, which promotes well-being and quality of life.

I. Chatzigiannakis et al. (Eds.): AmI 2019, LNCS 11912, pp. 170–185, 2019.
https://doi.org/10.1007/978-3-030-34255-5_12

Keywords: Cyber-physical systems · Adaptive acoustics · Internet of Things

1 Introduction and Motivation

This paper details the development of an acoustically adaptive modular system capable of enhancing Speech Clarity (C_{50} Clarity Index) in specific locations within a space in near real-time. It is designed and implemented as a sub-system within an open-ended and on-going development of an intelligent built-environment framework informed by both technical and architectural considerations. With respect to the technical, the present work is situated within the *Ambient Intelligence* (AmI) [1, 2]/*Ambient Assisted Living* [3]—or *Active and Assisted Living* [4, 5]—(AAL) discourse. With respect to the architectural, it is informed by the *Interactive Architecture* [6] and *Architectural Robotics* [7] discourses. The consideration of both aspects is central to the development of mutually complementary interoperability between physical and computational components within the built-environment.

The acoustically adaptive modular system is designed to improve the acoustic ambience via said enhancement of Speech Clarity and a concomitant noise-reduction in predetermined locations via mutually informing Physical/Mechanical and Computational components. Since sound is a potential environmental stressor associated with a variety of negative physiological, psychological, and cognitive responses [8], acoustic ambience is an important indicator of *Interior Environmental Quality* (IEQ) [9], which is strongly correlated with well-being and sustained quality of life [10]. The impact extends to a variety of programmatic functions as well as to specific spaces and audiences, not all partial to the context or character of AmI/AAL. For example, with respect to classrooms and children: Speech Clarity is strongly correlated with reading development among elementary school—i.e., second-grade—pupils [11]; and with respect to offices and adults: it is strongly correlated with intelligibility even in teleconference systems at the workplace [12], etc.

Although the scope of the detailed implementation consists of maximal C_{50} value-selection at octave band mid-frequency of 500 Hz, the same method and system may be used to select for C_{80}—Music Performance Clarity Index—or other acoustic features such as Reverberation Time, Definition (D_{50}), Early Decay Time, etc., at a variety of frequencies (e.g., 1000 Hz–8000 Hz). Accordingly, the present work is an instance-implementation of a method-type capable of enhancing acoustic ambience with respect to multiple acoustical parameters (in individual maximization or collective optimization). As with other sub-system developments belonging to the same open-ended intelligent built-environment framework, the present system is intended to operate intuitively, intelligently, and automatically in a closed-loop via inattentive or passive user-interaction yet without his/her intervention.

The system is presented in five sections. Section 2 describes the *Concept and Approach*, which explains the reasoning behind the physical as well as computational mechanisms and their interrelation. Section 3 details the *Methodology and Implementation*, which describes the actual implementation of a proof-of-concept fragment consisting of six modules. Section 4 presents the *Results*, which demonstrate the

successful operation of the system as corroborated by a performance overview. Finally, Sect. 5 provides a *Conclusion* and discusses present limitations and future work.

2 Concept and Approach

The present setup considers a virtual space of 4 m in length, 1.5 in width, and 1 in height—N.B.: the width and height correspond to the dimensions of the six-module fragment, and represent minimal dimensions for trials in the present setup. In this volume, the simulated sound-source is placed 3 m from the acoustical six-module fragment along the center-axis of the volume's length. The simulated receiver, and therefore the particular location where maximal C_{50} value is being selected for, is place at 1 m from the module fragment along the same axis. The virtual and physical acoustic modules measure 0.5 × 0.5 m (see Fig. 1; see Table 1 for *Absorption Coefficients*). The physical modules instantiate extension/retraction configurations corresponding to C_{50} calculations from this virtual space.

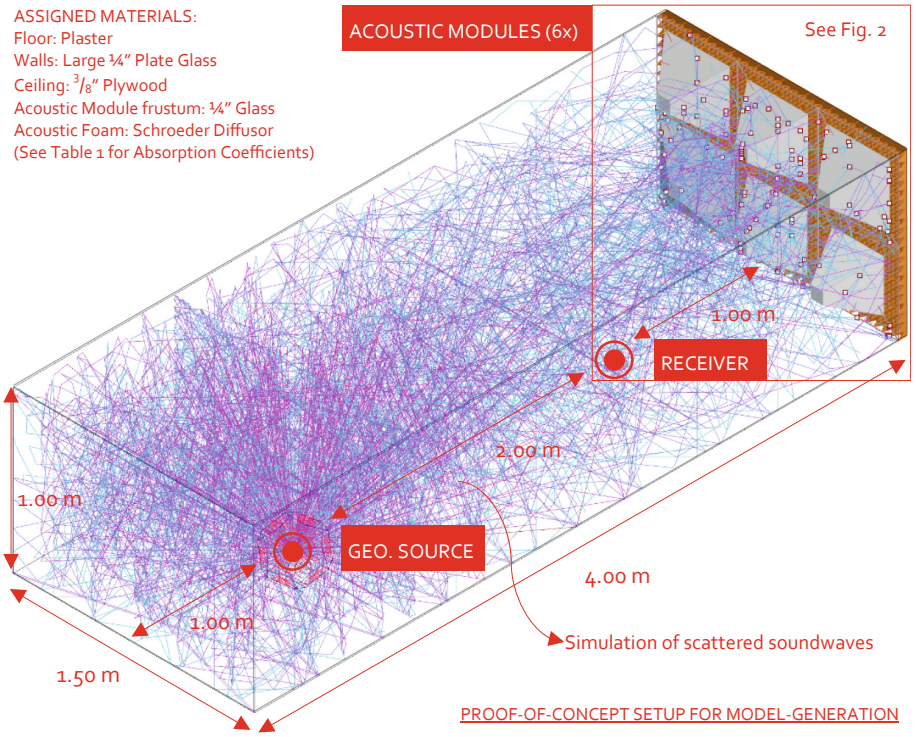

Fig. 1. Sample arrangement of six acoustically adaptive modules. Present configuration attains ~5.72 dB at a designated point in octave band mid-frequency of 500 Hz. Sound waves are represented in lines up to ten surface-bounces; the line colors are correlated with the sound-energy (i.e., darker to lighter equals more to less energy, respectively).

Table 1. Absorption coefficients (% energy absorbed)—from Pachyderm [16] material library

Hz	Plaster – rough on lath (*Floor*)	Large ¼" Plate Glass (*Pyramid Surfaces, Walls*)	$^{3}/_{8}$" Plywood Wall (*Ceiling*)	Schroeder Diffusor (*Acous. Absorber*)
62.5	2	25	32	18
125	2	18	28	22
250	3	6	22	24
500	4	4	17	32
1000	5	3	9	23
2000	4	2	10	19
4000	3	2	11	19
8000	2	2	13	19

The adaptive modular system is conceived as a *Cyber-Physical System* (CPS) [13, 14]. Its physical/mechanical component (see Sect. 3.1) consists of quadrilateral, truncated pyramidal modules that extend or retract perpendicularly to their base (see Figs. 2 and 4). This enables said modules (1) to change in the steepness of the sides of their frustum, which changes the way incoming sound waves are deflected/reflected/diffused by the surfaces of the pyramid; and (2) to reveal or to hide the absorbent material under each module, which enables a portion of incoming sound waves to be absorbed/dissipated in a controlled manner. The extension/retraction of each module—ranging from 70 mm to 270 mm in height—is controlled by the computational component of the system (see Sect. 3.2), which consists of two steps.

In the first, the initial extension/retraction of each module is determined by a generated model based on an evolutionary solver—viz., Galapagos [15] —selecting for maximal C_{50} values ascertained via an acoustical simulation software—viz., Pachyderm [16] —both running on Grasshopper [17]. In the second step, a simulated receiver at the location in question measures the clarity attained and updates the model's database with respect to the configuration's corresponding clarity-value. Since the nature of acoustics is not always exact, if the attained measurement is lower than the model's prediction for said location under the best module-configuration, but higher than the second-best configuration for the same location, the modules remain at the initial configuration. However, if the attained values are lower, this step reconfigures the modules to instantiate the second—or third-, fourth-, etc.—best configuration and updates the model's database with respect to the new optimal module-configuration value (see Fig. 3). These steps repeat each time the user moves to another location.

The computational model is an open-ended and closed-loop mechanism that is generated before any actual operation of the physical/mechanical component. It is open-ended in that, via its evolutionary solver (see Sect. 3.2), it continues to compute selected module extension/retraction configurations without a specific value as its selected target. It therefore continues to build its database of C_{50} values with respect to module extension/retraction configurations indefinitely. A caveat: the solver may be configured to end either after a particular period of time or when the difference between maximal values found becomes smaller than some relevant threshold—e.g., when the

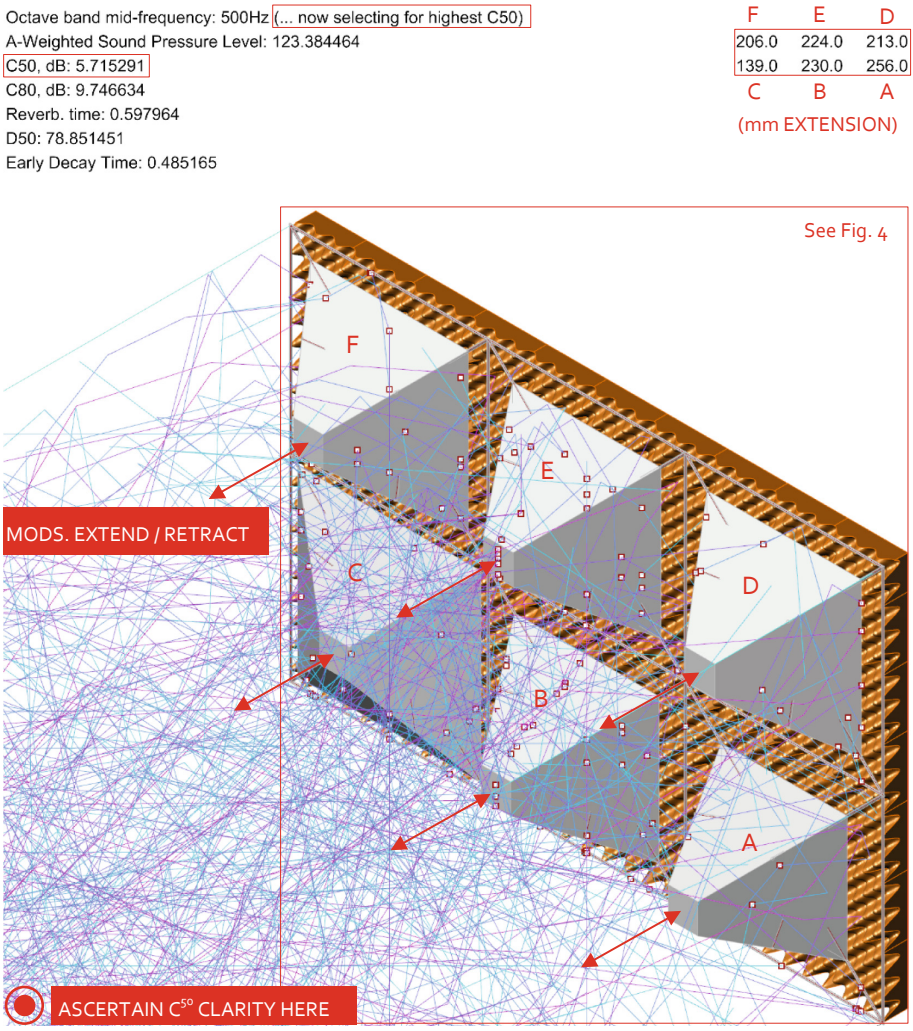

Fig. 2. Arrangement of six acoustically adaptive modules. Present configuration attains ~5.72 dB at a designated point in octave band mid-frequency of 500 Hz.

differences between values are found after ten decimal points, etc. Moreover, in cases where the *Fitness Landscape* or *Volume* (i.e., every fitness value resulting from the inter-combination of different variables or *genes*—see Rutten's discussion [18]) does have an actual optimum (either maximal or minimal value), the solver would end after having found it. However, this is not the case in the present system, as due to its complexity, there may be several equally satisfactory values whose difference is negligible (again, e.g., values with differences after ten decimal points).

The model is also closed-loop in that it updates its database from received feedback. That is, while the solver computes values from module extension/retraction

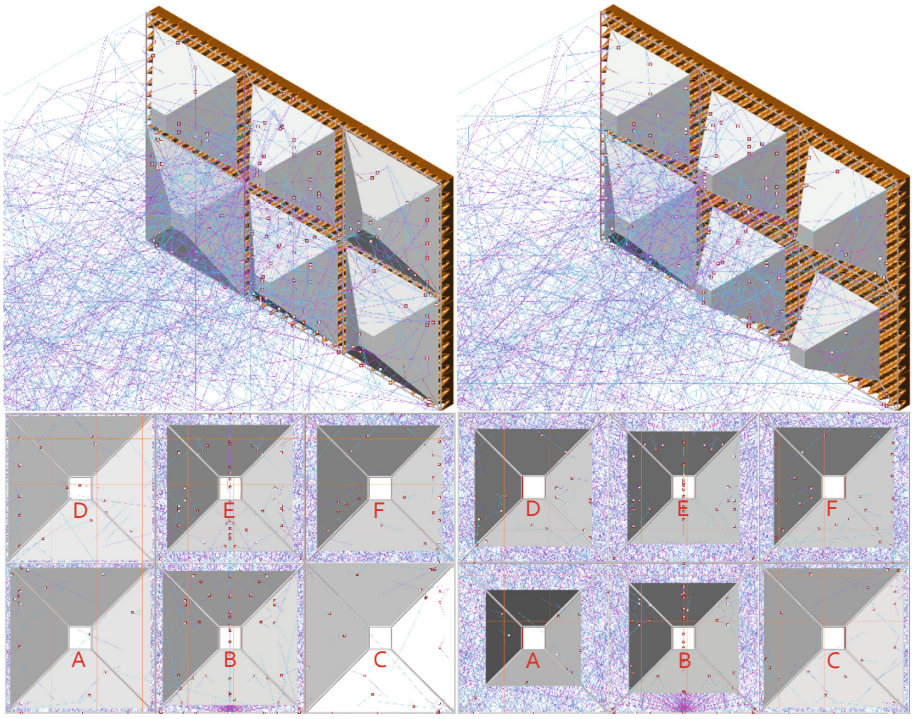

Fig. 3. Top-row: frontal isometric view. Bottom-row: posterior elevation view, viewing towards sound-source—N.B.: acoustic foam transparent). Left-column: iteration #20, with a C_{50} value of 3.8514 dB—acoustic module configurations (values in mm): A. 132; B. 159; C. 85; D. 112; E.187; F. 189 extension. Right-column: iteration #784, with a C_{50} value of 5.715291 dB—acoustic module configurations: A. 256; B. 230; C. 139; D. 213; E.224; F. 206 extension. Note that even small module configuration variations yield vastly different clarity values.

configurations and ranks them from highest (most optimal) to lowers (least optimal) indefinitely, the feedback mechanism updates the model's ranked database to reflect actual states of affairs—i.e., the actual supersedes the predicted/generated). In this manner, the model is constantly and continuously ascertaining the latest values and ranking them for use, which enables the physical/mechanical component to operate as soon as the model's database has at least one predicted/generated value. This is why a model extension/retraction configuration is said to be the most optimal with respect to maximizing C_{50} only up until the most recent iteration. Of course, an arguable minimum number of iterations (and therefore stored C_{50} values) is required in order to yield non-trivial module extension/retraction configuration suggestions—that is, to say that a particular module extension/retraction configuration is the optimal because there is only one generated/measured C_{50} value is to say nothing at all. While this setup risks yielding trivial results when the model's database is small, it also enables the system to potentially improve over time (see Sect. 5 for a caveat) with increasingly higher C_{50} values found in a more comprehensive database (for example, see Fig. 6 for a *Histogram* corresponding to presently computed C_{50} values).

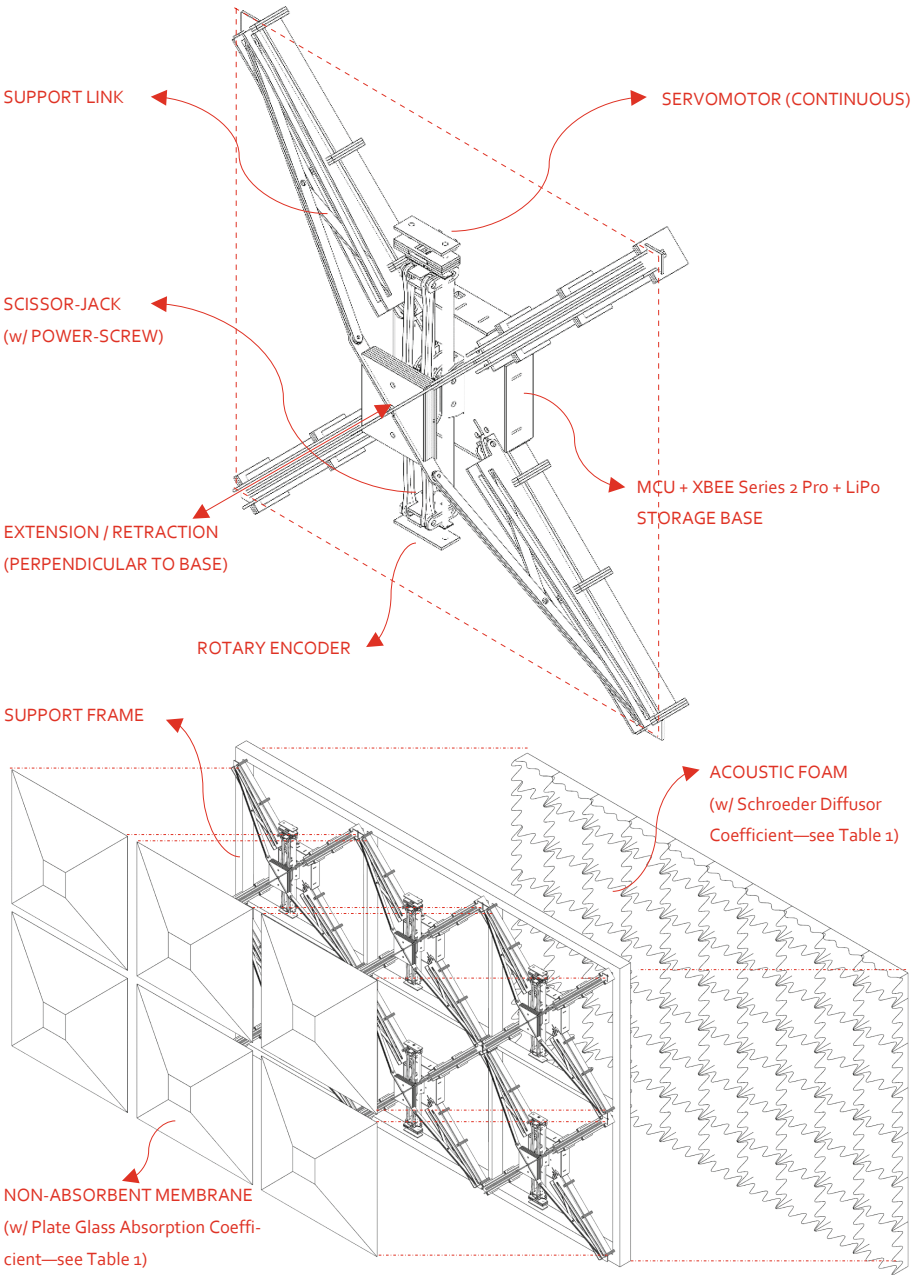

Fig. 4. Top: module breakdown. Bottom: implemented six-module fragment breakdown.

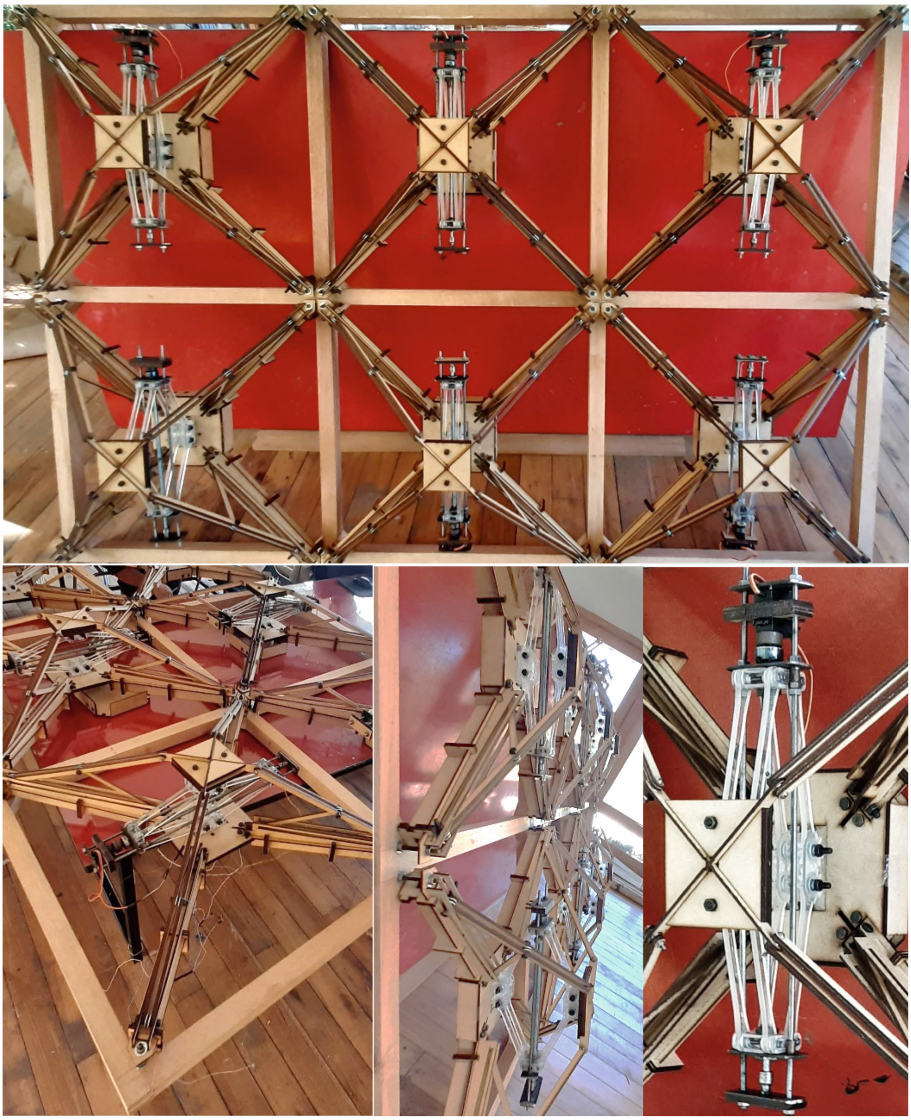

Fig. 5. Physical TRL-4/5 implementation of the six-module fragment used in laboratory tests.

3 Methodology and Implementation

The behavior of the physical acoustic modules with respect to their extension/retraction extents is determined by data gathered from the virtual space. Although the ascertained module configurations are expressed in the physical world, the real space is not correlated with the virtual one. That is to say, in the scope of the present implementation, the principal purpose is to demonstrate that a model configuration found via a virtual

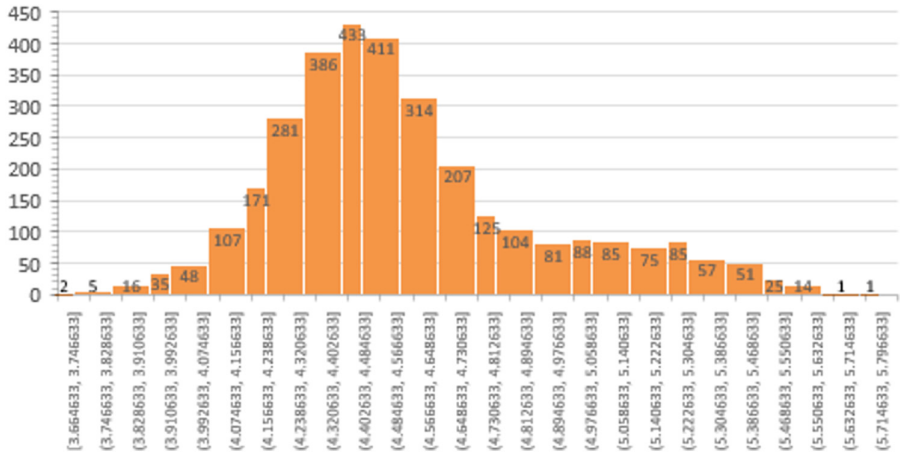

Fig. 6. Histogram of C_{50} values across 3,625 iterations generated in the present implementation.

space may be instantiated in the real-world; and that feedback corresponding to the real-world may be used to update the model's database. The particulars of both physical and computational components are detailed in the following sub-sections.

3.1 Physical/Mechanical Component: Acoustically Adaptive Modules

In terms of the physical build, each instance of the acoustically adaptive module is built with medium-density fiberboards, a steel tie-rod or power-screw, an acrylic scissor-jack as well as a variety of other metallic accessories such as ball-bearings, bolts, nuts, etc. These modules were designed as proof-of-concept instances appropriate to a *Technology Readiness Level* (TRL) [19] of 4–5 (see Fig. 5).

In terms of *Information and Communication Technologies* (ICTs), each module is equipped with a continuous servomotor (with 15 kg of torque) correlated with a rotary encoder to keep track of the rotation of the power-screw. The motor and encoder are connected to a *Microcontroller Unit* (MCU) attached at the base of the module. An XBee Series 2 Pro antenna as well as a LiPo battery are attached to the MCU—N.B.: While the modules depend on the main power supply to function, batteries are integrated as a contingency measure in case of power-outage. Each module serves as a node in a self-healing and meshed *Wireless Sensor and Actuator Network* (WSAN) whose System Architecture is inherited from previous developments by some of the authors [20]. Each node sends and receives data to and from any one of several coordinating devices built with *Raspberry Pi Zero W*s (RPiZWs). The computational model stores calculated C_{50} data in a shared database that the RPiZWs access via ZigBee in order to read module extension/retraction extents corresponding to each C_{50} value, which are then relayed—also in ZigBee—to each adaptive acoustic module for configuration instantiation. The present setup considers three coordinating RPiZWs as representative of a larger multitude in an intelligent built-environment's WSAN. This redundancy ensures system resilience, as operation would not be interrupted if one or

two coordinating nodes were to fail. Moreover, even if the selected communication protocol (i.e., ZigBee, for reduced energy-consumption) between the nodes were to fail, the sending and receiving of data and instructions would default to *User Datagram Protocol* (UDP) over WiFi. Redundancy of computational resources as well as communication protocols is essential to instantiate unobtrusive, intuitive, and independent data-driven intelligence. Finally, the LiPo battery is included as a secondary source of power, as the present implementation presupposes uninterrupted power availability from the main power supply.

3.2 Computational Component: Evolutionary Solver

As mentioned in Sect. 2, the computational component has two steps, where the first instructs physical modules to instantiate a particular extension/retraction configuration; and the second receives feedback and updates the model's database. Both steps are described in greater detail below. But it is worth emphasizing first that prior to either step, the spatial and material attributes and conditions of the virtual space must be assigned and/or determined. That is, the virtual space's (see Fig. 1) walls, ceiling, floor, etc., must be assigned material absorption attributes (see Table 1). Likewise, the sound-source and -receiver must also be explicitly specified and identified. From this, the acoustic simulation software—viz., Pachyderm [16] —may be used to compute a corresponding *Energy-Time Curve* (ETC) from which a variety of acoustical parameters may be ascertained. In the present case, C_{50} is the parameter of interest, which is an objective measure of Speech Clarity (vis-à-vis C_{80}, a measure of Music Performance Clarity) measured in *decibels* (dB). C_{50}, as a Clarity Index, reflects the fact that late sound-reflections degrade the intelligibility of speech due to the merging speech sounds. Similarly, very early sound-reflections, while not detrimental *per se*, will invariably contribute to intelligibility. The time-limit before sound-reflections become detrimental is agreed to be approximately 50 ms. C_{50} is calculated accordingly:

$$C_{50} = 10log_{10}\frac{\int_0^{50} p^2(t)dt}{\int_{50}^{\infty} p^2(t)dt}, \tag{1}$$

where $p(t)$ is the impulse-response sound-pressure at time t measured from direct-sound arrival [12]. The present implementation ascertains the highest C_{50} value possible in one octave band frequency: 500 Hz—although the same methods work for other bands.

In the first step of the computational component, the evolutionary solver—i.e., Galapagos [15] —instantiates an initial set of random module extension/retraction configuration in the virtual space and their corresponding C_{50} values are derived via the calculated ETC in Pachyderm [16]. The extension/retraction extents corresponding to the highest C_{50} gathered in this random set are relayed to the MCU of each corresponding module—respectively—and instantiated in the real world. While the physical instantiation takes place, the evolutionary solver continues to gather more C_{50} values from different configurations and ranks them from highest (most optimal) to lowest (least optimal). Only when a recent value is higher than the previously instantiated are new module extension/retraction extents relayed for physical instantiation. This means that given a specific location within a space, the module configurations will continue to

update in near-real time, always instantiating the most optimal value found thus far for that location. Since it is not the case that each iteration in the virtual space yields a progressively higher value, the module configuration will not be changing at each iteration. Instead, the most optimal module configuration thus far for that location will be instantiated automatically and in near real-time whenever the user steps in that location without the need of calculation—i.e., calculation takes place continuously in the background, at every iteration, but there is always a most optimal stored value corresponding to said location for fast physical instantiation.

As previously stated, the first set of module configurations is random. However, from that point onwards the extension/retraction extent of each module proceeds with evolutionary principles from information gathered from said random configurations. That is, each configuration contains an extension/retraction extent value for each module A, B, C, D, E, and F (see Fig. 2). This set or array of values per each configuration is called a *genome*. The *fitness*—in the present case: the C_{50} value—is determined for each *genome*, and the *Fitness Volume* is populated with these values. The fittest *genomes* are bred with one another to create a subsequent iteration or generation. As Rutten [18] points out, this breeding among the fittest *genomes* is necessary as it is unlikely that the initial randomized set instantiates the most optimal solution. Prior to the breeding process, those *genomes* deemed to be unfit—via a configuration option—are discarded and only the set of best performing pass their genes to the next generation. The offspring of the first randomized generation will have *genomes* whose *fitness* is distributed somewhere between that of their parents. A caveat cum limitation: in simpler systems—say, those with only two variables—it may be reasonable to assume that the offspring of fit parents will enjoy a relatively similar degree of fitness. However, this is not necessarily the case in the present six-variable system, where any one variable may have a negative impact over the positive selection of the whole. This *Fitness Volume* may be deemed chaotic due to its unpredictability. However, despite the individual chaos, rhyme and reason may still be derived when considering the sum average of all attained C_{50} values. That is, the *genomes* from each iteration/generation may yield *fitness* values not conforming to a discernible pattern, but the system may be said to be evolving positively if the sum average of said *fitness* values is said to be on the overall increase. In other words, the system is still evolving positively if, on average, subsequent generations are fitter than previous ones, which is the case in the present implementation (see Fig. 7)—see Sect. 5 for an expansion on this caveat with respect to limitations. Despite such potential limitations, evolutionary solvers are still considerably useful in problems that select for maximization or minimization of values. They are, for example, markedly more effective than trying all possible variable combinations in this six-variable system, which would require 64,000,000,000,000 calculations (i.e., the extension/retraction range of 200 mm of each module to the power of 6 modules).

In the second step of the computational component, after having physically instantiated a highest C_{50} value, an independent simulated receiver (also implemented with Pachyderm [16], yields a "measured" C_{50} value to compare with the model-computed value. Since the nature of acoustics is not always exact, simulated receivers may yield different C_{50} values under the same conditions. (N.B.: Even the same simulated receiver may yield a different C_{50} value if measured multiple times (see, for

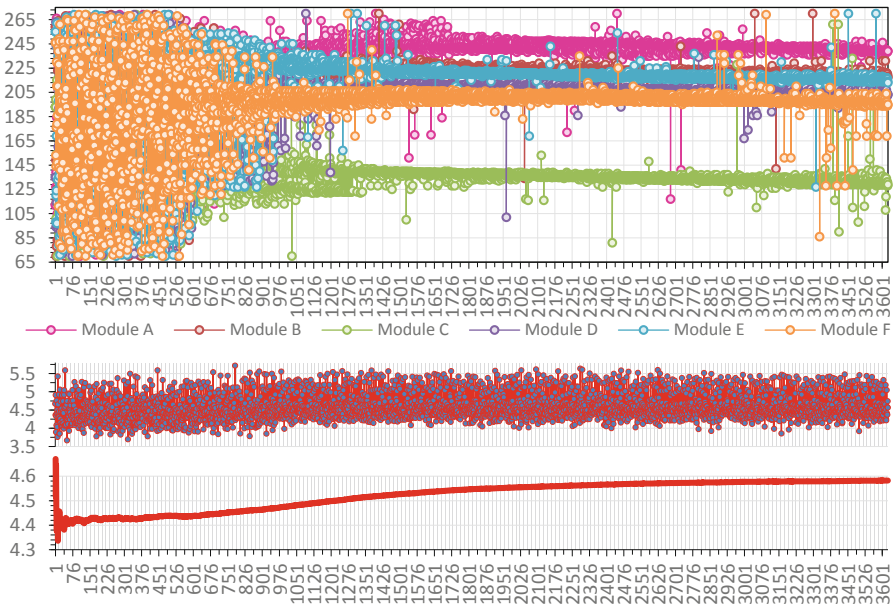

Fig. 7. Top: extension/retraction of each module with respect to each iteration. Middle: Corresponding C_{50} values for each *genome* with respect to each iteration. Bottom: Sum averages across all iterations.

example, iterations # 1 and 2 in Table 2, where with the same module extension/retraction configurations yielded 4.67 dB vs. 4.36 dB, respectively). But said values will also not be substantially different from each other. That is to say, the nature of acoustics is compatible with margins and tolerances.) If the independent receiver's C_{50} value is higher than that of the model's for that particular module configuration, the model's database is updated to reflect this independently measured value and the physical module configuration remains unchanged. If, however, the independent receiver's value is lower than the model's (i.e., lower than expected), then the system updates the model's database and proceeds to look for another module configuration that is expected to be higher than the independently measured value. That is to say, suppose that a given module configuration that corresponds to the highest ranked C_{50} value is expected to ascertain 5.00 dB at a given location but instead is independently measured to yield only 4.5 dB. At this point, the 5.00 dB in the model's database is replaced by 4.5 dB, and the system finds the second highest ranked C_{50} value and compares it to the 4.5 dB. If the second highest ranked C_{50} value is higher—say, 4.75 dB—then the system first proceeds to instantiate the module extension/retraction configuration that corresponds it and then to classify it as the new highest ranked C_{50} value. Hence what used to be the highest ranked C_{50} value is now the second (4.5 dB) and what used to be the second is now the first (4.75 dB). If, however, the second highest ranked C_{50} value is lower than the measured 4.5 dB value—say, 4.25 dB—then 4.5 dB and its corresponding module configuration remain highest ranked and

4.24 dB remains second highest. By virtue of the ranking mechanism, it is impossible for the third highest ranked to be higher than 4.25 dB, 4.5 dB, or 4.75 dB in this example. In other words, the second step of the computational component ensures that the independently measured C_{50} values are integrated back into the model's database always in a ranked manner, where the highest ranked is always the best module extension/retraction configuration to yield the highest C_{50} value for a specific location. This two-step process repeats at every configuration instantiation that yields a C_{50} value that is higher than the previous highest ranked value.

Table 2. Sample module configurations and resulting C_{50} values and sum averages.

Iter.	Mod. A (mm)	Mod. B (mm)	Mod. C (mm)	Mod. D (mm)	Mod. E (mm)	Mod. F (mm)	Output C_{50} (dB)	ΣAverage (dB)
1	112	259	193	135	255	246	4.670331	4.670331
2	112	259	193	135	255	246	4.362115	4.516223
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	132	159	85	112	187	189	3.8514	4.396181
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
784	256	230	139	213	224	206	5.715291	4.455165
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3625	239	218	129	203	213	195	4.563596	4.581994

4 Result

The computation component of the system executed 3,625 iterations in the present implementation. During this cycle, a maximum C_{50} value of ∼5.72 dB (at iteration # 784, see Table 2) and minimum of ∼3.66 dB are ascertained, with the sum average of all iterations being ∼4.58 dB. The difference between maximum and minimum is ∼2.05 dB.

The majority of module extension/retraction configurations yielded C_{50} values between ∼4.40 dB and ∼4.48 dB (see Fig. 6). This illustrates a salient advantage of implementing adaptive intelligence in the present system, as random manual operation would likely yield results in that range, which is lower than ∼5.72 dB. At 3,625 iterations it may be observed how each variable becomes increasingly attuned to a particular extension/retraction range (see Fig. 7, *Top*). However, as indicated by Fig. 7, *Middle*, it is difficult to see a pattern or correlation between this attunement and the resulting C_{50} values, which seemingly seem random. Nevertheless, as indicated by Fig. 7, *Bottom*, over time the sum average of the C_{50} values tend to increase, indicating that while individual *genomes* do not evidence positive evolution at each generation, the average of the collective does show important progress over time.

The physical modules shown in Fig. 5 instantiated highest ranked module extension/retraction configurations as expected, only instantiating a new configuration whenever a higher C_{50} value is found or when the user moves to another specific location in the real world as accounted for in the virtual space.

5 Conclusions

In this paper, the development of an acoustically adaptive modular system capable of enhancing Speech Clarity (C_{50} Clarity Index) in specific locations within a space in near real-time is presented. As a CPS, the system consists of a physical/mechanical and a computational component. The form and geometry of the modules conforming the mechanical component enable them (1) to change in the steepness of the sides of their frustum, which changes the way incoming sound waves are deflected/reflected/diffused by the surfaces of the pyramid; and (2) to reveal or to hide the absorbent material under each module, which enables a portion of incoming sound waves to be absorbed/dissipated in a controlled manner. The behavior of these modules is determined by two steps in the computational component of the system. First, the initial position of the modules is set via a model previously generated by an evolutionary solver [15], which identifies the optimal extension/retraction extent of each of the six modules to select for individual configurations that collectively ascertain the highest C_{50} value in said specific locations. Second, a simulated receiver at the location in question measures the actual clarity attained and updates the model's database with respect to the configuration's corresponding clarity-value. The system is conceived as a sub-system in an open-ended and on-going development of an intelligent built-environment framework, and as such it is conceived as one of many solutions designed to enhance well-being and quality of life in the context of AmI/AAL as well as *Interactive Architecture/Architectural Robotics* to instantiate environments that enable was Oosterhuis has called a *Society of Home, Society of Products, Society of Building Components* [21].

Although the system performed as expected, there are several limitations to consider, two of which are the most salient: (1) Sect. 2 mentions that the computational component of the system improves over time. This is true if and only if the selection process is not stuck in *local optima*—that is, a localized optimal value that is only said to be optimal with respect to a particular region of the *landscape* or *volume*. (2) Sect. 3.2 mentions that the system may be said to be evolving positively if subsequent generations are *fitter* than previous ones. This is indeed true, but the danger is that evolutionary solvers do not guarantee it. There may be instances where solutions are stuck in *local optima* and no further progress or positive evolution is observed after a certain iteration. This may be avoided by establishing or configuring mating and mutating-tolerances before running the evolutionary solver. However, it is not possible to eradicate the risk entirely. This makes a strong case for the generation of several models before committing to one. At present, work is being undertaken to implement the system via parallel computational models capable of not only updating within themselves but also across, thereby minimizing the risk of getting stuck in *local optima* by enabling the system to jump from one model to another and to select optimal C_{50} values from all available options.

Acknowledgements. The authors wish to acknowledge Cristian Amaguaña, Juan Balseca, and Dario Cabascango, students of the *Faculty of Electrical & Electronic Engineering* at *Escuela Politécnica Nacional*, for their assistance in the assembly of the physical implementation. Part of

the present implementation was made possible by funding from *Universidad Internacional SEK*'s Project No. P111819.

References

1. Kameas, A., Stathis, K. (eds.): AmI 2018. LNCS, vol. 11249. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03062-9
2. De Paz, J.F., Julián, V., Villarrubia, G., Marreiros, G., Novais, P. (eds.): ISAmI 2017. AISC, vol. 615. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61118-1
3. Calvaresi, D., Cesarini, D., Sernani, P., Marinoni, M., Dragoni, A.F., Sturm, A.: Exploring the ambient assisted living domain. A systematic review. J. Ambient Intell. Hum. Comput. **8**, 239–257 (2017)
4. Grzegorzek, M., Gertych, A., Aumayr, G., Piętka, E.: Trends in Active and Assisted Living - Open hardware architecture, Human Data Interpretation, intervention and assistance. Comput. Biol. Med. **95**, 234–235 (2018)
5. Flórez-Revuelta, F., Chaaraoui, A.A. (eds.): Active and Assisted Living: Technologies and Applications. The Institution of Engineering and Technology, Stevenage (2016)
6. Fox, M.: Interactive Architecture: Adaptive World. Princeton Architectural Press, New York (2016)
7. Green, K.E.: Architectural Robotics. Ecosystems of Bits, Bytes, and Biology. The MIT Press, Cambridge (2016)
8. Edelstein, E.A., Macagno, E.: Form follows function: bridging neuroscience and architecture. In: Rassia, S.T., Pardalos, P.M. (eds.) Sustainable Environmental Design in Architecture. Impacts on Health, pp. 27–42. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-0745-5_3
9. Roulet, C.-A., Bluyssen, P.M., Müller, B., de Oliveira Fernandes, E.: Design of healthy, comfortable, and energy-efficient buildings. In: Rassia, S.T., Pardalos, P.M. (eds.) Sustainable Environmental Design in Architecture. Impacts on Health, vol. 56, pp. 83–108. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-0745-5_6
10. Bluyssen, P.M.: The Healthy Indoor Environment. How to Assess Occupants' Wellbeing in Buildings. Routledge/Taylor & Francis Group, London (2014)
11. Puglisi, G.E., Prato, A., Sacco, T., Astolfi, A.: Influence of classroom acoustics on the reading speed. A case study on Italian second-graders. J. Acoust. Soc. Am. **144**, EL144 (2018)
12. Shimizu, T., Onaga, H.: Study on acoustic improvements by sound-absorbing panels and acoustical quality assessment of teleconference systems. Appl. Acoust. **139**, 101–112 (2018)
13. Serpanos, D.: The cyber-physical systems revolution. Computer **51**, 70–73 (2018)
14. Ochoa, S., Fortino, G., Di Fatta, G.: Cyber-physical systems, internet of things and big data. Future Gen. Comput. Syst. **75**, 82–84 (2017)
15. Rutten, D.: Galapagos. On the logic and limitations of generic solvers. Archit. Des. **83**, 132–135 (2013)
16. van der Harten, A.: Pachyderm acoustical simulation towards open-source sound analysis. Archit. Des. **83**, 138–139 (2013)
17. Grasshopper®: About Grasshopper. http://www.grasshopper3d.com/
18. Rutten, D.: Evolutionary Principles Applied to Problem Solving. Austria, Vienna (2010)

19. European Association of Research and Technology Organisations (EARTO): The TRL Scale as a Research & Innovation Policy TOOL. EARTO Recommendations. http://www.earto.eu/fileadmin/content/03_Publications/The_TRL_Scale_as_a_R_I_Policy_Tool_-_EARTO_Recommendations_-_Final.pdf
20. Bier, H., Liu Cheng, A., Mostafavi, S., Anton, A., Bodea, S.: Robotic building as integration of design-to-robotic-production and -operation. In: Bier, H. (ed.) Robotic Building, 1. Springer International Publishing AG (2018). https://doi.org/10.1007/978-3-319-70866-9_5
21. Oosterhuis, K.: Caught in the act. In: Kretzer, M., Hovestadt, L. (eds.) ALIVE. Advancements in Adaptive Architecture, vol. 8, pp. 114–119. Birkhäuser, Basel/Berlin/Boston (2014)