

A Matlab-CONTAM Toolbox for Contaminant Event Monitoring in Intelligent Buildings

Michalis Michaelides, Demetrios Eliades, Marinos Christodoulou, Marios Kyriakou, Christos Panayiotou, Marios Polycarpou

► **To cite this version:**

Michalis Michaelides, Demetrios Eliades, Marinos Christodoulou, Marios Kyriakou, Christos Panayiotou, et al. A Matlab-CONTAM Toolbox for Contaminant Event Monitoring in Intelligent Buildings. Harris Papadopoulos; Andreas S. Andreou; Lazaros Iliadis; Ilias Maglogiannis. 9th Artificial Intelligence Applications and Innovations (AIAI), Sep 2013, Paphos, Greece. Springer, IFIP Advances in Information and Communication Technology, AICT-412, pp.605-614, 2013, Artificial Intelligence Applications and Innovations. <10.1007/978-3-642-41142-7_61>. <hal-01459653>

HAL Id: hal-01459653

<https://hal.inria.fr/hal-01459653>

Submitted on 7 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Matlab-CONTAM Toolbox for Contaminant Event Monitoring in Intelligent Buildings

Michalis P. Michaelides^{1,2*}, Demetrios G. Eliades², Marinos Christodoulou²,
Marios Kyriakou², Christos Panayiotou², and Marios Polycarpou²

¹ Department of Electrical Engineering and Information Technologies
Cyprus University of Technology

30 Archbishop Kyprianos Str., CY-3036 Lemesos, Cyprus

² KIOS Research Center for Intelligent Systems and Networks, and
Department of Electrical and Computer Engineering
University of Cyprus

75 Kallipoleos Ave., CY-1678 Nicosia, Cyprus

Abstract. An intelligent building should take all the necessary steps to provide protection against the dispersion of contaminants from sources (events) inside the building which can compromise the indoor air quality and influence the occupants' comfort, health, productivity and safety. Multi-zone models and software, such as CONTAM, have been widely used in building environmental studies for predicting airflows and the resulting contaminant transport. This paper describes a developed Matlab Toolbox that allows the creation of data sets from running multiple scenarios using CONTAM by varying the different problem parameters. The Matlab-CONTAM Toolbox is an expandable research tool which facilitates the implementation of various algorithms related to contamination event monitoring. In particular, this paper describes the implementation of state-of-the-art algorithms for detecting and isolating a contaminant source. The use of the Toolbox is demonstrated through a building case-study. The Matlab-CONTAM Toolbox is released under an open-source licence, and is available at <https://github.com/KIOS-Research/matlab-contam-toolbox>.

Keywords: Intelligent buildings, multi-zone model, CONTAM, Matlab Toolbox, contaminant event monitoring, multiple scenarios, detection, isolation.

1 Introduction

An Intelligent Building is a system that incorporates computer technology to autonomously govern and adapt the building environment in order to enhance operational and energy efficiency, cost effectiveness, improve users' comfort, productivity and safety, and increase system robustness and reliability [1, 5]. The

* Corresponding author: michalis.michaelides@cut.ac.cy.

dispersion of contaminants from sources (events) inside a building can compromise the indoor air quality and influence the occupants' comfort, health, productivity and safety. These events could be the result of an accident, faulty equipment or a planned attack. Distributed sensor networks have been widely used in buildings to monitor indoor environmental conditions such as air temperature, humidity and contaminant concentrations. Real-time collected data can be used to alert occupants and/or control environmental conditions. Accurate and prompt identification of contaminant sources can help determine appropriate control solutions such as: (i) indicating safe rescue pathways and/or refugee spaces, (ii) isolating contaminated spaces and (iii) cleaning contaminant spaces by removing sources, ventilating and filtering air. Therefore, the accurate and prompt identification of contaminant sources should be an essential part of the Intelligent Building design.

To study the security-related problems in intelligent buildings, large quantities of data are required to be simulated under various conditions, in order to capture the variations in the complex dynamics involved. For the creation of non-stationary datasets related to the presence of contaminants in intelligent buildings, the Matlab-CONTAM Toolbox has been developed and released, which is the main contribution of this work. In the Toolbox, we utilize the computational engine of CONTAM [7], a multi-zone simulation software developed by the US National Institute of Standards and Technology (NIST). With CONTAM, the user can easily create the building outline and specify the zone volumes, the leakage path information and the contaminant sources present. This information can be further utilized for calculating the air-flows and resulting contaminant concentrations in the various building zones. A limitation of the CONTAM v3.1 software is that it can only analyze a single scenario at a time. Furthermore, no algorithms are included within CONTAM for detecting and isolating contaminant sources.

The Matlab-CONTAM Toolbox features a user-friendly Graphical User Interface (GUI) and a modular architecture. It allows the creation of multiple scenarios by varying the different problem parameters (wind direction, wind speed, leakage path openings, source magnitude, evolution rate and onset time) as well as the storage of the computed results in data structures. The data from these scenarios are further analyzed by the developed algorithms for determining solutions for contaminant event monitoring. In this paper, we implement inside the Toolbox, state-of-the-art algorithms for detecting and isolating a contaminant source in the indoor building environment. These are further demonstrated using a 14-zone building case study referred to as the Holmes' house. We should point out that the presented case study is only a small sample of the Toolbox's possibilities. Using the Toolbox, the user can easily create data sets for any building scenario by choosing which parameters to vary. The user can then select an algorithm to analyze these data sets. A key idea behind this work is to provide a software that enables the application of computational intelligence methods in buildings' related research [2–4, 9]. This issue is particularly relevant

and challenging at the same time since it is rather difficult to define and propose benchmarks and/or testbeds for learning in non-stationary environments.

The rest of the paper is organized as follows. First, in Section 2, we describe the architecture and the main functionality of the developed Matlab-CONTAM Toolbox. Section 3 describes the model and the implemented algorithms for contaminant event monitoring in intelligent buildings. Then, in Section 4, we demonstrate the Toolbox and how it is applied for the contaminant source detection and isolation problem. Finally, Section 5 provides some concluding remarks and presents our plans for future work.

2 Matlab-CONTAM Toolbox Architecture

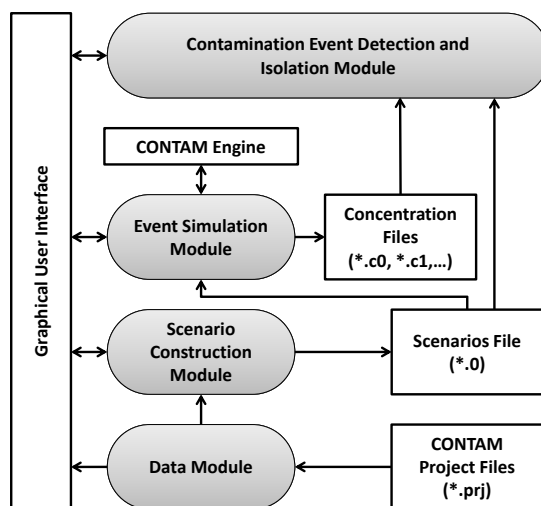


Fig. 1. The software architecture of the Matlab-CONTAM Toolbox.

The Matlab-CONTAM Toolbox Architecture is depicted in Fig. 1. It features a modular design and a user-friendly Graphical User Interface (GUI). The “Data Module” opens CONTAM Project files (*.prj) and reads the information related to the building parameters. These include the building outline, the zone volumes and the leakage path information. The building zone schematics are extracted from this information and used by the GUI of the Toolbox to plot the building.

The information from the “Data Module” are then used by the “Scenario Construction Module”. This module is responsible for specifying the parameters which are to be considered to construct one or multiple simulation scenarios, through the GUI. These include environmental parameters that affect the flow dynamics (i.e. wind speed and direction, temperature, opening status of doors

and windows), as well as contaminant source parameters (i.e. magnitude, evolution rate and onset time). The scenarios along with building information are stored in the Scenarios File (*.0). The data stored in the Scenarios File are used by the “Event Simulation Module”, which communicates with the CONTAM engine to compute a state matrix for each different flow scenario. When the state matrices have been constructed, the module simulates the different contamination scenarios and computes the contaminant concentration at each zone. All the results are stored in one or more Concentration Files (*.c0, *.c1, ...).

The “Contamination Event Detection and Isolation Module” is comprised of state-of-the-art algorithms for detecting whether a contamination event has occurred in the building, and isolating its zone. The module uses information from the Concentration Files and the Scenarios Files, as computed by the Scenario Construction Module and the Event Simulation Module. The Contamination Event Detection and Isolation Module communicates with the GUI to configure the parameters of the algorithms and to view the results.

3 Contaminant Event Detection and Isolation

In this section, we provide an overview of the theoretical results used in the development of the “Contamination Event Detection and Isolation Module.” Next, we describe the state-space model and the implemented algorithms for Contaminant Detection and Isolation (CDI) in intelligent buildings. Let \mathbb{R} represent the set of real numbers and $\mathbb{B} = \{1, 0\}$ the set of binary. Using the multi-zone modeling methodology, the state-space equations for contaminant dispersion in an indoor building environment with n zones can be presented in the following general form,

$$\begin{aligned} \dot{x}(t) &= (A + \Delta A)x(t) + Q^{-1}Bu(t) + Q^{-1}Gg(t) \\ y(t) &= Cx(t) + w(t), \end{aligned} \quad (1)$$

where, $x \in \mathbb{R}^n$ represents the concentrations of the contaminant in the building zones, while $A \in \mathbb{R}^{n \times n}$ is the state transition matrix which models changes in contaminant concentration between the different building zones primarily as a result of the air-flows. The term ΔA collectively accounts for the presence of modeling uncertainty in the building envelope as a result of changing wind speed, wind direction and variable leakage openings. Through the Toolbox it is possible to characterize and calculate some bounds on this uncertainty, as we will be demonstrating in Section 4. The controllable inputs in the form of doors, windows, fans and air handling units are represented by $u \in \mathbb{R}^p$ while $B \in \mathbb{B}^{n \times p}$ is a zone index matrix concerning their locations. The final term of the first state-space equation involves the location and evolution characteristics of the contamination sources represented by $G \in \mathbb{B}^{n \times s}$ and $g \in \mathbb{R}^s$ respectively. Note that $Q \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the volumes of the zones, i.e. $Q = \text{diag}(Q_1, Q_2, \dots, Q_n)$ where Q_i is the volume of i -th zone. In the second equation, $y \in \mathbb{R}^m$ represents the sensor measurements, $C \in \mathbb{B}^{m \times n}$ is a zone index matrix for the sensor locations and $w \in \mathbb{R}^m$ stands for additive measurement noise.

More details on the state-space formulation can be found in [6]. Note that a similar state-space formulation has been commonly used by the fault diagnosis community to represent models of uncertain non-linear systems [10]. In this context, the final term in (1), i.e. $\mathcal{Q}^{-1}Gg(t)$, is used to describe process faults that disturb the normal system operation.

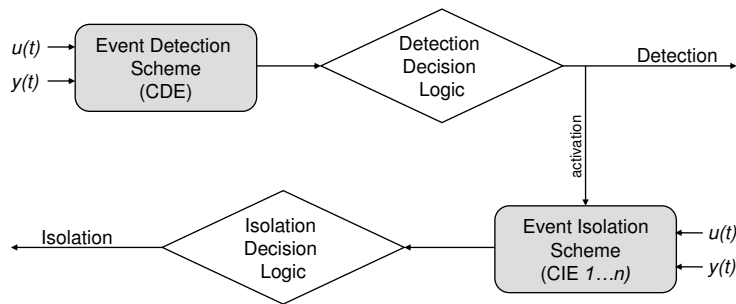


Fig. 2. Architecture of detection and isolation of a single contaminant source.

The general architecture of the implemented CDI algorithms using state estimation techniques is shown in Fig. 3. For event detection, the Contaminant Detection Estimator (CDE) estimates the contaminant concentration in the different building zones under the “no source” hypothesis. It then compares the actual contaminant concentration as measured by the sensors to the estimated one. Event detection is decided if the difference between actual and estimated concentration (residual) of at least one sensor exceeds the prescribed adaptive threshold. Following detection, a bank of n isolators is activated, one for each building zone. The j -th Contaminant Isolation Estimator (CIE), $j \in \{1, \dots, n\}$, estimates the contaminant concentrations in the j th building zone under the “source present” hypothesis. Then, it compares the actual contaminant concentrations as measured by the sensors to the estimated ones. If the residual for the j -th CIE exceeds the threshold, then zone j is *excluded* from being a possible candidate for the source location. Isolation is decided after $n - 1$ zones have been excluded, leaving only one possible building zone containing the source. More details on the implemented algorithms including the derivation of the adaptive thresholds can be found in [10].

4 Case study

In this section, we demonstrate the proposed Matlab-CONTAM Toolbox for detecting and isolating a contaminant source in an indoor building environment

through a case study corresponding to the Holmes' house experiment [8]. The interface of the Toolbox is depicted in Fig. 4, loaded with the project file corresponding to the Holmes' house. As shown in the figure, the building is comprised of 14 zones: a garage (Z1), a storage room (Z2), a utility room (Z3), a living room (Z4), a kitchen (Z5), two bathrooms (Z6 and Z13), a corridor (Z8), three bedrooms (Z7, Z9 and Z14) and three closets (Z10, Z11 and Z12). There are a total of 30 leakage path openings corresponding to windows and doors (P1–P30). It is assumed that natural ventilation is the dominant cause of air movement in the building with wind coming from the east (90°) at a speed of 10 m/s. All the openings (doors or windows) are assumed to be in the fully open position. We assume that at time 3 hours, a contaminant source of generation rate 126.6 g/hr is activated in the utility room (Z3) as shown in Fig. 4. There is one sensor in each zone able to record the concentration of the contaminant at regular intervals at its own location but the sensor measurements are corrupted by noise. Based on the sensor measurements, our goal is to detect and isolate the source under conditions of noise and modeling uncertainty.

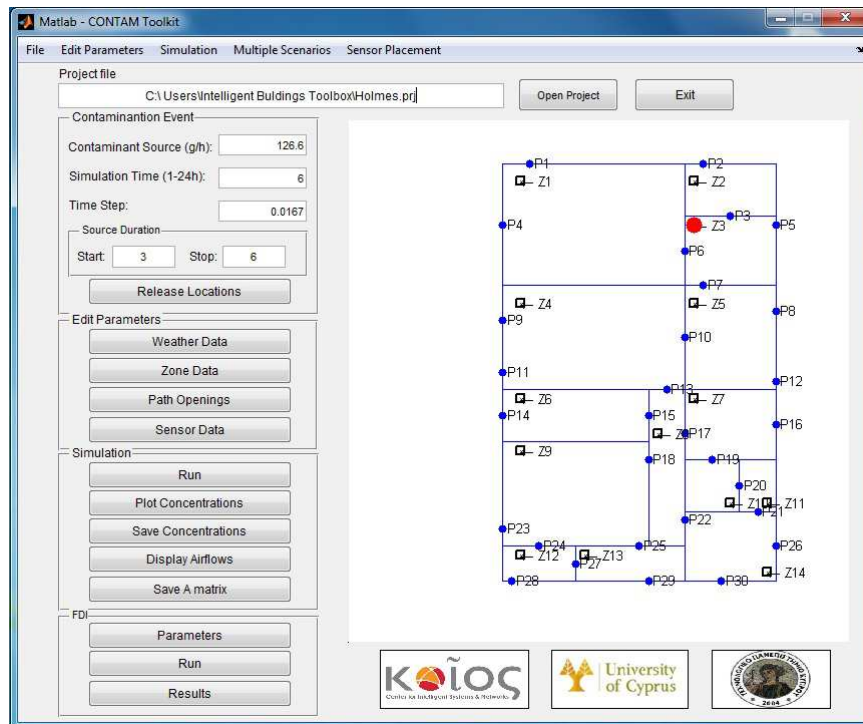


Fig. 3. The Matlab-CONTAM Toolbox displaying the Holmes' House project.

The main body of the Matlab-CONTAM Toolbox GUI is divided into four main sections as shown in Fig. 4: Contamination Event, Edit Parameters, Simulation and CDI. In the *Contamination Event* section, the user can specify the contamination source(s) characteristics including the release location(s), the generation rate, the onset time and the duration of the event. From this section, it is also possible to choose a simulation time between 1-24 hr with the appropriate time step. The various problem parameters can be modified in the *Edit Parameters* section, including the Weather Data, the Zone Data, the Path Openings and the Sensor Data. In the *Simulation* section, the user can run the contaminant transport simulation once all the parameters are appropriately set. This calls CONTAM in the background for calculating the airflows as explained in Section 2, and activates the other options inside the *Simulation* section. From here, the user can view and save the time series of contaminant concentrations in the different building zones (which have sensors installed), display the airflows and save the state transition matrix corresponding to these flows for future reference.

In the *CDI* section, the user can change the parameters, run and view the results for the event detection and isolation algorithms outlined in Section 3. The *Parameters* option displays another interface as shown in Fig. 4, divided into four sub-sections: Uncertainties bound, Detection and Isolation Parameters, Noise Bound and Nominal A matrix. The *Uncertainties bound* sub-section is used to provide a bound on the modeling uncertainty ΔA in model (1). This can be specified in three different ways using the provided dialog box: (i) as a constant known value; (ii) calculated using random sampling within the specified tolerances around the nominal conditions on the wind direction, wind speed, temperature and path openings; and (iii) calculated from an external file (*.mat) of a set of state transition matrices. For example, in the interface depicted in Fig. 4, the bound on the modeling uncertainty is calculated as 0.237 using the second option with 100 random samples within the following intervals from the nominal conditions: wind direction $90 \pm 10^\circ$, wind speed $10 \pm 0,5$ m/s, zone temperature $30 \pm 2^\circ\text{C}$ and leakage openings $\pm 10\%$ from the fully open position. In the *Detection and Isolation Parameters* sub-section, the user can modify, if required, the default values used by the detection and isolation algorithms displayed in Fig. 4. These depend on the initial problem assumptions and include the initial maximum state estimation error for detection and isolation, the learning rate and the maximum interval value for estimating the contaminant source, and an initialization value for the estimated source. More information on these parameters can be found in [10]. Next, a bound on the noise needs to be specified in the *Noise Bound* sub-section. Currently, we are assuming uniform, bounded noise, but other types of noise (i.e. Gaussian) can easily be incorporated in the future. Finally, the nominal state matrix A in (1) can be used in its current form or loaded from a file using the *Nominal A matrix* sub-section.

After setting the various parameters, the *Run* option is used in order to run the CDI algorithms and view the results. The sequence of steps performed follows the CDI architecture outlined in Section 3. The user is notified for the progress of the detection and isolation algorithms through user friendly graphical

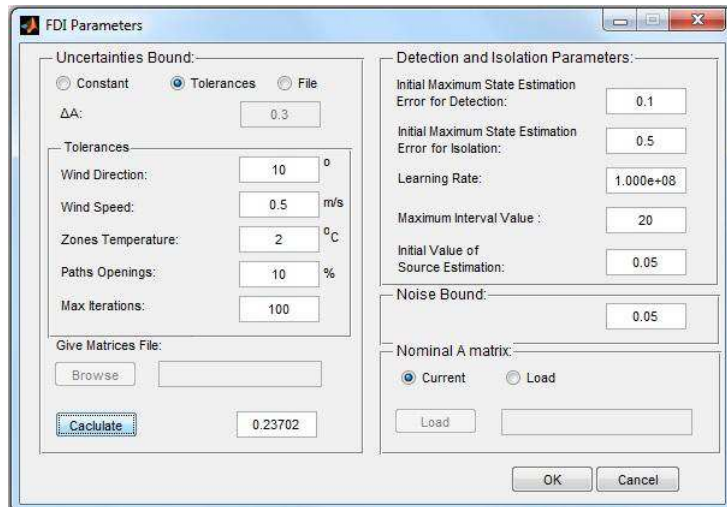


Fig. 4. Interface for setting the CDI parameters and calculating bound on modeling uncertainty.

displays and message notification banners. Following the completion of the CDI algorithms, the user can view more detailed results concerning detection and isolation, by using the *Results* option. This opens up a new interface which displays the results of detection and isolation. For the specific test case scenario, the source was detected 3 *min* after the release time and isolated 7 *min* after the release time in the Utility zone. The *Results* interface is fully configurable and allows the user to plot any result concerning the residuals and/or the adaptive thresholds used for detection (CDE) and isolation (CIEs) for the zones of his choice. Some indicative plots of the results are displayed in Fig. 4–4.

5 Conclusions and Future Work

Contaminant event monitoring in intelligent buildings requires large quantities of simulation data under different conditions in order to capture the variations in the complex dynamics involved. To this aim, the Matlab-CONTAM Toolbox has been developed and released, a software that operates within the Matlab environment and provides a programming interface for CONTAM, a multi-zone airflow and contaminant transport analysis tool. To facilitate the interaction of the researchers with the Toolbox, an intuitive graphical user interface has been designed for accessing the different functionalities and a modular architecture has been adopted to assist in the implementation of new methods. In this paper, we describe the implementation of advanced CDI schemes based on the state-space method for detecting and isolating a single contaminant source in an indoor

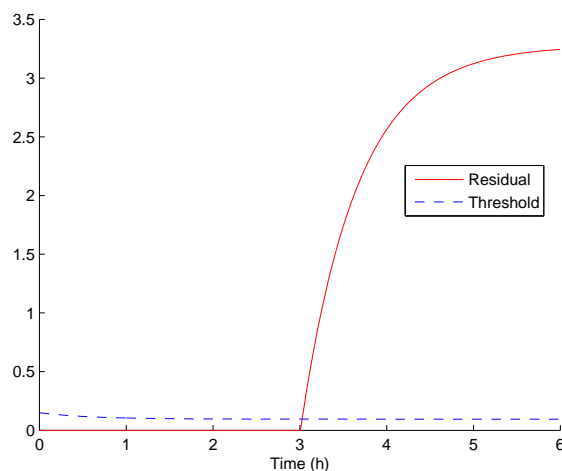


Fig. 5. Contaminant Detection Estimator (CDE) for zone 3 (i.e. sensor in utility room). The output estimation error (residual) is displayed using a solid line while the adaptive threshold is displayed using a dashed line. The contaminant source is detected when the error exceeds the threshold 3 min after the release.

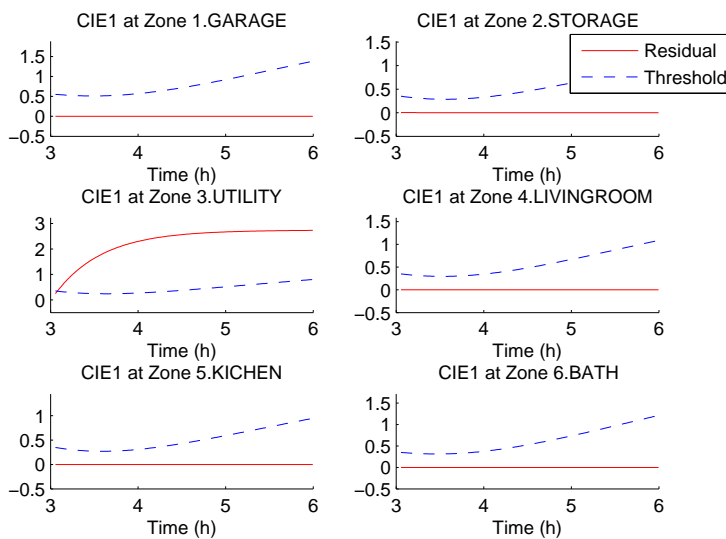


Fig. 6. 1st Contaminant Isolation Estimator (CIE) for zones 1–6. The output estimation error (residual) is displayed using solid lines while the adaptive thresholds are displayed using dashed lines. Zone 1 is excluded as a possible source location when the error exceeds the threshold in zone 3.

building environment. We demonstrate the use of the Toolbox through a specific 14-zone building case study referred to as the Holmes' house.

In the future, we plan to expand the capabilities of the Toolbox for detecting and isolating multiple contaminant sources. We also plan to incorporate different techniques for event detection. The ultimate goal of the Toolbox is to serve as a common programming framework for research on Intelligent Buildings, by allowing the simulation of multiple contamination scenarios under varying conditions as well as the storage of the results in data structures. These data, can serve as benchmarks in the future for evaluating the performance of the different designed algorithms.

Acknowledgements This research work has been funded by the European Commission 7th Framework Program, under grant INSFO-ICT-270428 (iSense), and by the Cyprus Research Promotion Foundation's Framework Programme for Research, Technological Development and Innovation, co-funded by the Republic of Cyprus and the European Regional Development Fund.

References

1. Braun, J.: Intelligent building systems-past, present, and future. In: Proc. of American Control Conference (ACC'07). pp. 4374–4381. IEEE (2007)
2. Dounis, A., Caraiacos, C.: Advanced control systems engineering for energy and comfort management in a building environment—a review. *Renewable and Sustainable Energy Reviews* 13(6–7), 1246 – 1261 (2009)
3. Eliades, D., Michaelides, M., Panayiotou, C., Polycarpou, M.: Security-oriented sensor placement in intelligent buildings. *Building and Environment* 63(0), 114 – 121 (2013)
4. Fong, K., Hanby, V., Chow, T.: HVAC system optimization for energy management by evolutionary programming. *Energy and Buildings* 38(3), 220 – 231 (2006)
5. Lu, X., Clements-Croome, D., Viljanen, M.: Past, present and future of mathematical models for buildings. *Intelligent Buildings International* 1, 23–38(16) (2009)
6. Michaelides, M., Reppa, V., Panayiotou, C., Polycarpou, M.M.: Contaminant event monitoring in intelligent buildings using a multi-zone formulation. In: Proc. of Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'12). vol. 8, pp. 492–497 (2012)
7. Walton, G., Dols, W.: CONTAM 2.4 user guide and program documentation. National Institute of Standards and Technology, NISTIR 7251 (2005)
8. Wang, L., Dols, W., Chen, Q.: Using CFD capabilities of CONTAM 3.0 for simulating airflow and contaminant transport in and around buildings. *HVAC&R Research* 16(6), 749–763 (2010)
9. Wong, S., Wan, K.K., Lam, T.N.: Artificial neural networks for energy analysis of office buildings with daylighting. *Applied Energy* 87(2), 551 – 557 (2010)
10. Zhang, X., Polycarpou, M.M., Parisini, T.: Design and analysis of a fault isolation scheme for a class of uncertain nonlinear systems. *Annual Reviews in Control* 32(1), 107–121 (2008)