Chapter 1

Bayesian Networks for Supporting Model Based Predictive Control of Smart Buildings

Alessandro Carbonari, Massimo Vaccarini and Alberto Giretti

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/58470

1. Introduction

Optimal behaviour is one the most desired features of contemporary technological systems. Challenges like secure operation, energy efficiency, and reliable performance call for the optimised behaviour of any systems that operate and interact in our living environment. The challenge in achieving optimised performances resides in the uncertainty that qualifies the environment surrounding technical systems. Whatever model drives the systems' behaviour, it must be able to face unforeseen events, to manage the vagueness of the sensing apparatus and the errors of the control devices. Bayesian statistics is one of the theoretical backgrounds that support the construction of systems which are able to act effectively inside complex environments. Bayesian statistics is grounded on the fundamental premise that all uncertainties should be represented and measured by probabilities. Then, the laws of probabilities apply to produce probabilistic inferences about any quantity, or collection of quantities, of interest. Bayesian inference can provide predictions about probability values pertaining time series or can model parameters in terms of probability distributions that represent and summarize current uncertain knowledge and beliefs. Bayesian inference uses a kind of direct causal or model-based knowledge to provide the crucial robustness needed to make the optimised behaviour of technical systems feasible in the real world [1]. Once this kind of models have been built, then theoretically sound evidence propagation algorithms are used to update the belief set about the external environment and about the system performance, on the basis of acquired evidence. This is the fundamental mechanism that drives the construction and the operation of intelligent systems based on Bayesian inference. This chapter describes a sample engineering application of this approach on a large scale. It concerns the design and the development of an intelligent building energy management system (smart BEMS) that is able

to optimise the operation of the mechanical air supply systems of the Passeig De Gracia metro station in Barcelona. To the purpose of this application, predictive models were developed to support the optimal control of environmental conditions in the station, which was necessary due to the many interacting variables of the domain.

Building Energy Management Systems (BEMSs) are control systems installed in buildings for managing the building's mechanical and electrical equipment, such as ventilation, lighting, fire and security systems [2]. BEMSs consist of hardware and software components. The hardware set-up of a BEMS is typically made up of sensor-actuator networks that accurately monitor the indoor-outdoor environment and the building plants state. The software side of a BEMS consists of a number of functional layers that implement standard management functionalities like plant status monitoring, alarm management, demand driven plant management, reporting, etc. The hardware side of the commercial BEMS technology is at present a rather mature field. A number of initiatives and associations both at industrial and public level (e.g. European Building Automation and Controls Association-EU.BAC) are cooperating to develop open communication and seamless integration standards such as BACnet, KNX, LonWorks [3], and DALI [4]. The software side of commercial BEMSs is being standardised as well. Standard EN15232 provides a structured list of controls, building automation and technical building management functions that make an impact on the energy performances of buildings. Firstly, it provides a method to define the minimum requirements concerning the building automation and the building management policies, differentiated according to the level of complexity of buildings; secondly, it provides detailed methods to assess the impact of these policies on the energy performance of any given building. Nevertheless, EN15232 methods are limited to relatively simplified applications, ranging from simple homeostatic control to demand-driven and time-scheduled policies. The implementation of optimised control policies that encompass the complex weather and end-user dynamics in the energy management of buildings is still missing. The analysis of standard BEMS applications suggests that only a fraction of the available BEMS energy saving potential of each specific building is utilized by the implemented management policies, thus missing significant opportunities for reducing operating costs through better supervisory controls. Frequently, plant and building set-points follow prescribed schedules and are not optimized in response to changing dynamic conditions, including weather, internal loads, occupancy patterns, etc. Nonetheless, there are significant opportunities for optimizing control set points and modes of operation in response to dynamic forcing functions and utility rate incentives. A number of studies [5-8] have shown potential savings for optimized controls in the range of 10% to 40% of the overall cooling cost.

Model Predictive Control (MPC) [9-11] may be used to enhance BEMSs so that they can improve their control performances getting close to optimal behaviour. MPC is an advanced control technique [12] that uses the predictions of future building status, obtained by means of a model of the building's dynamics, in order to solve the problem of determining the optimal control policies. The purpose of building management is to guarantee comfort at minimum operational cost. The MPC integrated approach to building management guarantees performance over the full range of conditions which are likely to be encountered. Since the predictions

http://dx.doi.org/10.5772/58470

that serve the optimal control are obtained through model simulation of the building future states, the implementation of MPC requires the development of integrated models capable of predicting the near future behaviour of the controlled environment under specific conditions, so that the optimal solution can be sought through scenario analysis [13, 14]. Adaptive and predictive control strategies would follow from these considerations. The development of domain models that are able to drive the MPC of a complex set of systems, like the ones operating in a metro station, is not a trivial task. For example, on one side MPC models must provide accurate predictions of future states but, on the other side, they must be computationally light in order to provide predictions in a time frame compatible with the monitoring time constraints. Furthermore, MPC models must interoperate with real sensor/actuator networks that usually, for cost reasons, cannot be larger than few tenths of devices and whose deployment is constrained by a number of external factors. Nevertheless, the model accuracy must be granted despite the reduced representation of the physical model and the suboptimal selection of the parameter set. The fulfilment of such competing requirements compels the definition of a modelling framework that, by guiding the MPC modeller through a set of methodological steps, will contribute to design accurate and robust models, which are sufficiently light to be embedded in real control systems. The Bayesian inference approach, and its computational counterpart Bayesian Networks, provide the means to manage the requirements set that compels the development of MPC models.

This chapter will outline the Bayesian Network approach that was followed to develop the environmental model used to design the line 3 of "Passeig De Gracia" (PdG-L3) metro station energy control system and the main features of the hybrid modelling solution undertaken to fulfil its functional requirements. A metro station is a very complex system. It involves, among others, multi-storey underground spaces having multifaceted thermal behaviour, e.g., intricate air exchange dynamics with the outside, heat conduction with the surrounding soil and high variable internal gains due to travelling passengers and trains. Furthermore, a metro station is usually serviced by various equipment involving cooling, ventilation, safety and security, lighting, vertical transportation and horizontal passenger transfer, gates and selling machines, information and auxiliary systems. The research illustrated in this paper is one of the results of the SEAM4US project funded under EU grant n. FP7-2011-NMP-ENV-ENERGY-ICT-EeB-285408. The objective of the SEAM4US research is to develop an advanced control system for the PdG metro station in Barcelona capable of dynamically adjusting the internal environment in optimal way, based on forecasts regarding the external environment, in order to guarantee energy efficiency, comfort and comply with regulations.

2. Literature review about MPC control

As mentioned in the Introduction, the Bayesian networks developed in this chapter were used to provide forecasts about the future state of the PdG-L3 in Barcelona, given the knowledge about their current state, in order to support the application of a Model based Predictive Control (MPC) approach. In fact, MPC is an enhancement of adaptive control.

It is known that any control in buildings is targeted to minimize power consumption while keeping required comfort level and guaranteeing robustness of the solution. In order to fit these specifications, the control system must comply with several features. It must be optimal, i.e. it finds out the values of a vector of design parameters that yield optimal system performance evaluated by a so-called cost function. In addition, the control system must be adaptive, which is "a special type of nonlinear control system which can alter its parameters to adapt to a changing environment. The changes in environment can represent variations in process dynamics or changes in the characteristics of the disturbances. […]" [15]. Robustness is also required, thus implying that the models used for designing the controller should consider all process dynamics and must be able to adapt to unknown conditions. Finally, the predictive feature is another opportunity for achieving high energy efficiencies: prediction gives the capability of taking soft control actions in advance instead of suddenly reacting to unexpected deviations from the required state, thus saving energy.

MPC works based on a model of the building dynamics and the solution of an optimization problem to determine the optimal control inputs. It takes into account the (measured) current state of the system, future weather conditions and other disturbances (e.g. internal gains), in order to control actuators (e.g. HVAC, lighting and blind systems), so that energy and money usage are minimized. At the current point in time, a heating/cooling plan is formulated for the next several hours to days, based on predictions of the upcoming weather conditions. The control action is designed by running the model of the process over a given prediction horizon and evaluating the control sequence that gives the minimum value of the cost function [16]. Based on the results from this computation, the first step of the control policy is applied to the building, setting all the HVAC components, before moving one step forward and repeating the process at the next sampling time. This receding horizon approach is what introduces feedback into the system, since the new optimal control problem solved at the next time will be a function of the new state at that point in time, and hence of any disturbances that have meanwhile acted on the building. The final result will be a trajectory of inputs and states into the future that satisfy the dynamics and constraints of the system while optimizing some given criteria.

One remarkable survey about the effectiveness of MPC was carried out by means of simulations and applied to office buildings [17]. First, the authors considered and compared a list of potential adaptive approaches, among which we cite reduction of the thermal comfort when the building is not used, widening of the room temperature comfort range, use of Indoor Air Quality controlled ventilation. Results coming from these approaches were then compared with the benefits of a model based predictive control. The building was simulated by means of a single zone, twelfth order, time discrete bilinear building model of coupled thermal, air quality and light dynamics [18, 19]. Those preliminary simulations showed that the highest energy savings were determined by predictive control, which was also the best one at reducing the number of comfort violations encountered during the process.

What makes the application of predictive control to the “Passeig de Gracia” (PdG-L3) metro station in Barcelona very meaningful is that this is the case of a large underground building where the interaction with the outdoors is very critical and it can be modelled using very

http://dx.doi.org/10.5772/58470

complex analytical approaches and occupancy figures result someway difficult to predict. Hence, the dynamics of the station cannot be solved –and predicted– though a simplified thermal model (e.g. of statistical type); so, the reported work focuses on the key problem regarding the development of predictive models relative to complex domains, which was faced through the adoption of Bayesian Networks. In fact, they gave back a lumped representation of a number of sub-systems, involving thousands of variables.

The overall MPC control framework applied to the station is represented in Fig. 1. Inputs u to the system are the variables that can be driven by the controller (e.g. frequency that drives injector fans in the case of mechanical air supply). The outputs y are the power consumption and indicators for comfort and health that must be controlled in order to reach certain desired reference level r. The relation between inputs and outputs is also significantly affected by a set of disturbances d, such as weather, train arrival, passenger flows and fans external to the station: they cannot be manipulated but only "accounted for" by using direct measures, whenever possible, together with a disturbance model. At each control step, the prediction model receives candidate input sequences $\hat{u}$ picked out by the controller; disturbance predictions come from disturbances models $\hat{d}$, measured outputs m from PdG-L3 and the prediction model estimates the future output sequence $\hat{y}$. The optimal control sequence u^* is that one which minimizes a given cost function while complying with given constraints. Once the optimization problem has been solved, the first step u of the optimal sequence is applied as the best control action. The overall procedure is repeated at each step, thus closing the control loop. The implementation of those systems asks for the development of devices and services:

1. monitoring systems and intelligent algorithms to interpret occupant's behaviour [20];
2. high-level control systems capable of solving optimization problems in real-time;
3. accurate and fast dynamic models of buildings' behaviour and their systems, necessary to feed the high-level control systems (i.e. to generate the predicted output sequence $\hat{y}$);
4. accurate modelling of disturbances (e.g. occupancy, weather conditions etc..).

The predictive models mentioned by bullet no. 3 above were developed in the form of Bayesian Networks, because they were able to simulate the complexity of the system under analysis while keeping the computational effort manageable for real-time applications. To this aim Section 4 presents the basics on probability inference and Bayesian learning, despite the fact that this chapter cannot cover all the algorithms related to these topics. In Section 5 the indices useful to evaluate the quality of the developed networks were shown. In the same section the problem of network instantiation which includes even uncertainty is unfolded. Section 6 will report the procedure used to develop the Bayesian Networks object of this chapter, the validation of their inference capabilities and the cost function implemented by the controller to search the optimum solutions. Finally, in Section 7 the networks were wrapped within the whole predictive modelling framework and their capabilities shown through one example. Conclusions are given in Section 8.

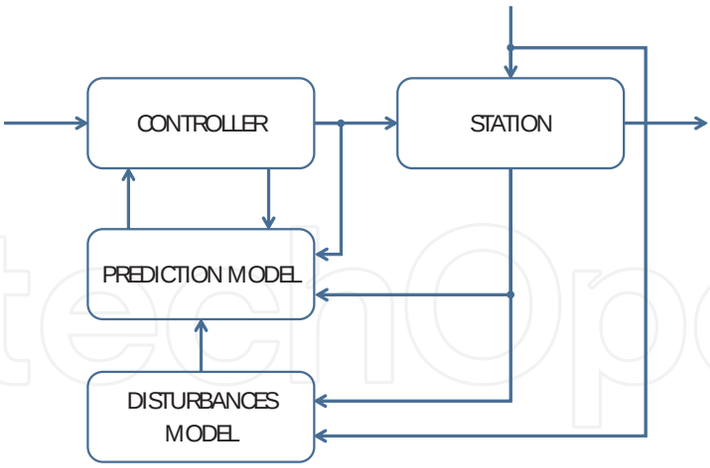

Figure 1. Predictive model based control framework defined for the metro station PdG-L3.

3. Basics of Bayesian Networks

3.1. Inference propagation

According to the framework outlined in Section 2, the current state of a building will be monitored in real-time by means of a sensor network deployed inside it. However, in order to limit the number of sensors to be installed in buildings, only a fraction of all the relevant physical variables are measured, while the remaining variables are derived from models. In other words, these models allow to estimate indirect measurements from measurements made directly by sensors. In the case study we are presenting in this Chapter, these models were developed in the form of Bayesian Networks, mainly because they are suitable to reduce complex domains into computationally manageable models. This is a key feature when computations must be performed in real-time. While numerical models (e.g. the Dymola™ model mentioned in Section 6) take hours to simulate the fluid dynamics and thermal behaviour of the PdG station, Bayesian networks make computations in a matter of seconds or minutes. Hence they are the most suitable to perform real-time predictions, provided that a reliable procedure for their development is available.

Other features typical of Bayesian Networks, which might come of advantage in these applications, are their capability of managing incomplete (e.g. one or a few data are not available because the corresponding sensors are broken) and uncertain information (e.g. if we include uncertainty in sensor measurements or if inputs are relative to forecasts of disturbance actions).

Whenever a Bayesian Network estimates indirect measurements from direct measurements, in fact it implements inference algorithms. Such inferences are computationally possible, thanks to the conditional probability relationships defined among the variables of the domain under analysis. This allows to consider just the most relevant relationships among all the variables (i.e. nodes), which are kept in the form of conditioned probabilities of the kind

http://dx.doi.org/10.5772/58470

$P(X_2|X_1)$, where X_1 is a parent of X_2 (which is its child node instead) and X_2 is conditionally independent of any variable in the domain which is not its parent [21]. The "chain rule" descends from this concept, in that the joint probability of a group of variables in a domain can be determined by the knowledge of the state of just its parents, thus limiting the database required for its inference [22]:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}) \quad (1)$$

More remarkably, Bayesian Networks allow to perform inferences, in other words any node can be conditioned upon new evidences, even when they are relative to multiple variables. This feature is particularly important in case a control system must work in real-time, because in that case evidences acquired about a state variable (i.e. from sensor measurements) must be propagated to update the state of the rest of the domain. This process requires conditioning, and it might be called also probability propagation or belief updating. It is performed via a flow of information throughout the network, without limitation in the number of nodes [1]. When it is run in the MPC framework, the controller will make queries to a set of nodes belonging to the networks, whose probability distributions are computed from the state of other nodes, upon which observations (or evidences) are already available (e.g. the future state of disturbance variables and the current state of the physical domain). To this purpose the Bayes theorem is exploited when there is a need to reverse the inference. In particular, if inference is run from causes to consequences it is called predictive reasoning; otherwise, if inference is directed from consequences to causes, it is called diagnostic reasoning. Inference in Bayesian Networks is solved by complex combinations of algorithms [1]. In order to show how this works in the case of BEMs, a short example will be discussed. The first step towards the development of any Bayesian Network is defining its graphic structure, which requires all the variables of the domain to be ordered and causal relationships among them to be defined. The three elementary structures used to order variables in Bayesian Networks are: causal chains, causal tress and poly-trees (Fig. 2). Then, other more complex structures may be formed as a combination or enhancement of these elementary fragments. The computational burden would change as a consequence.

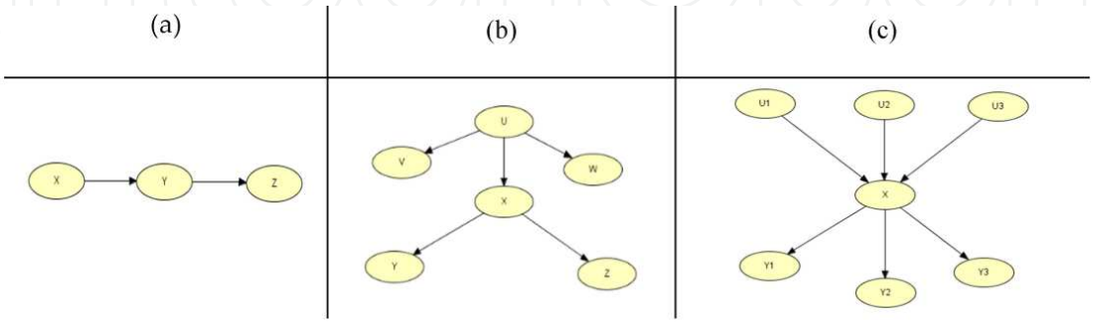

Figure 2. Graphic representation of a causal chain made up of three nodes (a), a causal tree (b) and a causal poly-tree (c).

Just to provide an example, the first structure depicted in Fig. 2-a could be useful to represent the case of sun radiation hitting and heating up the external surface of an envelope, as a consequence rising its temperature and making heat flux towards the interior. In case thermal heat gains cannot be directly measured, an alternative indirect estimation can be made: first we measure sun radiation hitting the external surface of the wall (e.g. by means of a solar pyranometer), then a model of the envelope is developed in the form of a Bayesian Network, which estimates in real-time heat gains by means of inference propagation. These models would get sun radiation intensity measured by the pyranometer as their input, then inference would infer the most likely value of internal heat gains (Fig. 3). Such an indirect estimation needs belief propagation (or probabilistic inference) based on dedicated algorithms. The notation in Fig. 3 corresponds to the notation in Fig. 2-a, provided that X stands for "sun radiation", Y stands for "wall temperature" and Z stands for "heat gain". In the latter figure four states were defined for each random variable (or event) and all the probability values were assumed as a uniform probability function (i.e. all their states are equally likely), because no learning had been done. The states of the variables were separated into intervals, according to the knowledge learned about the physical system by the model's developer, who assumed sun radiation to be limited between 0 and 400 W/m^2, wall temperature between 0 and 65°C, heat gains between 0 and 4.5 W/m^2. Once probability learning is performed as explained in the next sub-section, the network can be used to evaluate how evidence about the first variable (i.e. "sun radiation") is propagated towards the rightmost variable ("heat gain") passing through the intermediate one ("wall temperature").

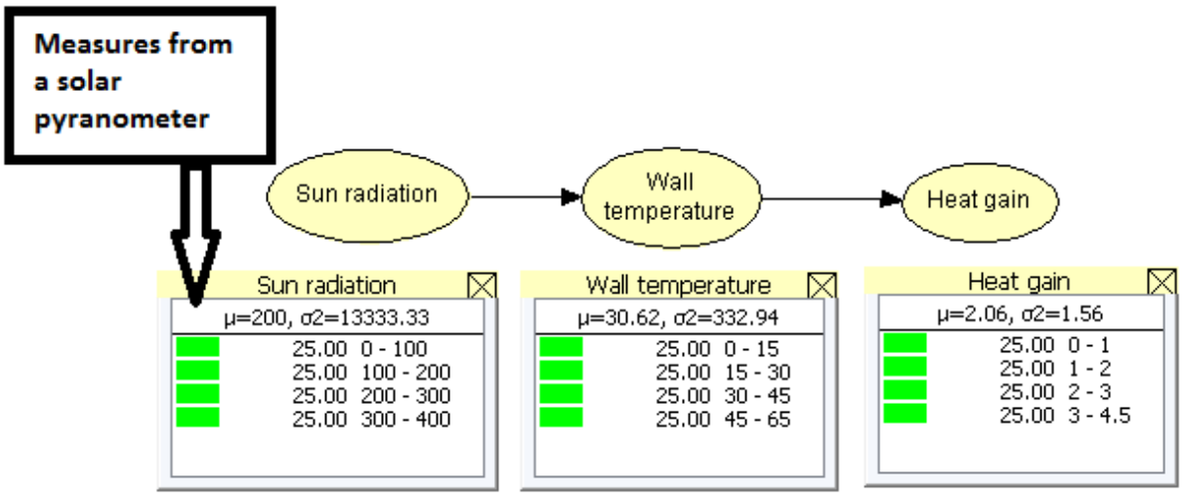

Figure 3. Example where a Bayesian chain can be used to infer indirect measurements (heat gains) from direct measurements (sun radiation intensity).

In order to make inference propagation feasible, a conditional probability table must be defined for each couple of variables in the chain. The model-based knowledge embedded in any Bayesian Network is represented graphically by a directed link between any variable X (e.g. "sun radiation") and Y (e.g. "wall temperature"), which is quantified by a fixed conditional probability matrix:

http://dx.doi.org/10.5772/58470

$$M_{Y|X} = P(y \mid x) = \begin{bmatrix} P(y_1 \mid x_1) & P(y_2 \mid x_1) & \dots & P(y_n \mid x_1) \\ \dots & \dots & \dots & \dots \\ P(y_1 \mid x_m) & P(y_2 \mid x_m) & \dots & P(y_n \mid x_m) \end{bmatrix} \quad (2)$$

where y_i and x_i are the generic states of variable nodes Y and X, respectively. In the case in Fig. 3 both X and Y might occur in one of the four possible states. Inference propagation needs to know first the probability assigned to every state of the second node Y conditioned to each state of the first node X. A more comprehensive notation is given by the form $BEL(x)$, which reflects the overall belief accorded to proposition $X=x$ by all evidence so far received, hence $BEL(x)=P(x \mid \xi)$. Hence this represents the set of dynamic values obtained as a result of the updated belief accorded to proposition $X=x$ once all the evidences about its parents is collected in the event ξ. In the simplest Bayesian topology, which is represented by "chains", if evidence $\xi=\{Y=y\}$ is observed, then from Bayes theorem the belief distribution of X (diagnostic reasoning) is given by:

$$BEL\ (x) = P(x \mid \xi) = \frac{P(x) \bullet P(\xi \mid x)}{P(\xi)} = \beta \bullet P(x) \bullet \lambda(x) \quad (3)$$

where the likelihood vector is given by:

$$\lambda(x) = P(\xi \mid x) = P(Y = y \mid x) = M_{y|x} \quad (4)$$

As a consequence, the likelihood of x is the y's column of the matrix in eq. (2). It is stored as an input in node Y, so that it can be transmitted as a message to X, thus enabling X to compute its belief distribution $BEL(x)$, having as many states as the number of rows in matrix $M_{y|x}$.

Propagation is possible even if Y is not observed directly but is supported by indirect observation $\xi=\{Z=z\}$ of a descendant Z of Y, which means the chain is of the kind in Fig. 2-a. It can still be written:

$$BEL\ (x) = P(x \mid \xi) = \beta \bullet P(x) \bullet \lambda(x) \quad (5)$$

Then, conditioning and summing upon the values of Y:

$$\lambda(x) = P(\xi \mid x) = \sum_y P(\xi \mid y,\ x) \bullet P(y \mid x) = \sum_y P(\xi \mid y) \bullet P(y \mid x) = M_{y|x} \bullet \lambda(y) \quad (6)$$

where the fact that Y separates X from Z was used. Due to eq. (4), $\lambda(y)=P(\xi \mid Y)=M_{z|y}$, so in this case it is derived from the conditional probability matrix between variables Y and Z. To the purpose of conditional updating in the chain, the state of X is irrelevant to infer the state of Z, once the state of Y is known, which might be explained by the sentence "Y d-separates X from Z" [1].

Starting from these concepts, further and more complex computations can be accomplished, so as to propagate inference throughout networks with any kind of admissible connections, but they all need the prior definition of conditional probability tables between linked variables.

3.2. Learning conditional probability tables

In the case of PdG-L3 presented in this chapter, the Bayesian Networks were built in the Hugin™ software environment. This software offers three approaches to learn the entries of conditional probability tables of the kind in eq. (2): from subjective knowledge; from analytical equations; from datasets. As the procedure presented in paragraph 6 learned conditional probability tables from datasets put together through numerical simulations, this is the case that will be considered in this sub-section. In particular, the algorithm used by Hugin™ is called "EM learning". In Fig. 4 we depicted two examples of conditional probability tables required by Hugin™ in order to perform the inference propagations explained in sub-section 4.1. The left sided one (Fig. 4-a) is the table needed to define how node *Y* of the network in Fig. 2-a is conditionally dependent to node *X*; while the right sided one (Fig. 4-b) is the table needed to define the likelihood of each state of the node *X* with respect to the joint combinations of the states of nodes *U1*, *U2* and *U3* (i.e. *X*'s parents). It is worth remarking that in this second case the number of columns is equal to the number of combinations of states of the nodes which are parents of *X*.

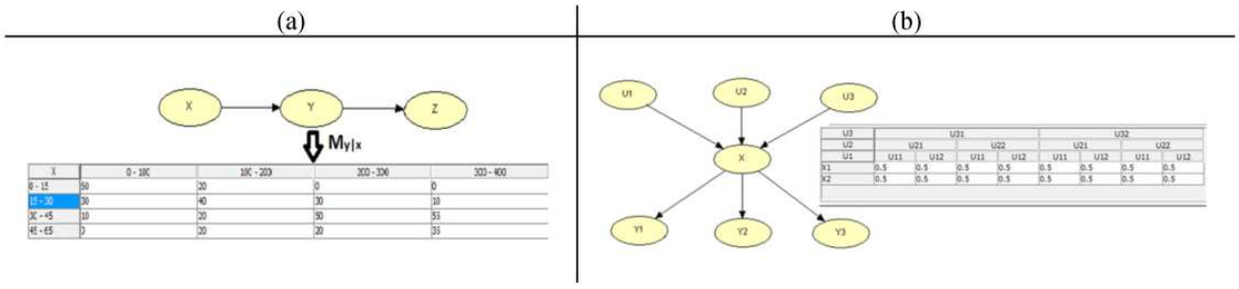

Figure 4. Conditional probability tables relative to the a node having just one parent with four states (a) and to a node having three parents with two states each (b).

The case we are considering might be described as a set of discrete variables $U=\{x_1, \ldots, x_n\}$, whose multivariate joint probability distribution can be encoded in some particular Bayesian Network structure B_s. The structure of B_s for our case study was derived from expert knowledge, hence just probabilities must be inferred from the additional knowledge provided by a random sample. Also, and according to the references in this field, it is assumed that the random sample $D=\{C_1, \ldots, C_m\}$ contains no missing data [23], which means that each case C_l consists of the observations of all the variables in U. Also, we can say that D is a random sample from B_s. Indeed, B_s may be thought as a directed acyclic graph that encodes assertions of conditional independence. In fact, it orders the variables in domain U such that the joint probability distribution can be estimated by means of the chain rule in eq. (1). Now, for every X_i there will be some subset $\Pi_i \subseteq \{X_1, \ldots, X_n\}$ such that X_i and $\{X_1, \ldots, X_n\}$ are conditionally independent given Π_i. That is:

$$P(X_i \mid X_1, \ldots, X_n) = P(X_i \mid \Pi_i) \tag{7}$$

As a consequence, a Bayesian Network is made up of a set of local conditional probability distributions and a set of assertions of conditional independence. These conditional independences can be described like in eq. (7), where the parents of X_i are grouped in the set Π_i, and are useful to "d-separate" any variable (i.e., to make it conditionally independent) from the rest of B_s.

Given B_s, let r_i be the number of states of variable X_i; and let $q_i = \prod_{x_l \in \Pi_i} r_l$ be the number of states of Π_i. Let θ_{ijk} denote the physical probability of $X_i=k$ given $\Pi_i=j$ for $i=1,..,n$, $j=1,..,q_i$, $k=1,..,r_i$. Adopting this notation, the following equivalences are valid [23]:

$$\vartheta_{ij} \equiv \cup_{i=1}^{n} \{\vartheta_{ijk}\} \quad \vartheta_{Bs} \equiv \cup_{i=1}^{n} \cup_{j=1}^{q_i} \{\vartheta_{ij}\}, \tag{8}$$

In other words, eq. (8a) states that the two notations are equivalent and represent the case where all the physical probabilities of $x_i=k$ are grouped, once any x_i in domain U is selected and the states of its parents is fixed at any $\Pi_i = j$. As a consequence, eq. (8b) represents all the physical probabilities of the joint space B_s (i.e. the Bayesian Network structure), because it encompasses all the states of Π_i (i.e., parents of x_i) and all the variables x_i in domain U. Where "$\cup \ldots$" stands for the union of all the states represented by that expression. The probability distribution of child nodes must be described by a probability distribution, which may be then updated according to evidence acquired by its parents. The software Hugin™ uses the EM algorithm, which defines a Dirichlet distribution for each variable θ_{ij} (i.e. one distribution for any variable of B_s given its parents are in the state j) [23]:

$$p(\vartheta_{ij} \mid B_s, \xi) = c \bullet \prod_{k=1}^{r_i} \vartheta_{ijk}^{N'_{ijk}-1} \tag{9}$$

where c is a normalization constant (in the form of a gamma function), N'_{ijk} is the multinomial parameters of that distribution, limited between 0 and 1, finally ξ is the observed evidence. The Dirichlet distribution describes the probability of one of the variables x_i when it varies all over its states. One of the advantages provided by using a Dirichlet distribution is that it is completely defined by its multinomial parameters. In addition, its shape can be easily adapted to fit various probability density function. Finally, and more importantly, it can be demonstrated that the learning process is made easier in this way [23]. In fact, the values N'_{ijk} represent expert knowledge introduced in the network by the developer themselves. Then, if N_{ijk} is the number of observations in the database D, in which $x_i=k$ and $\Pi_i=j$, we are able to update that distribution by adding empirical information meant by the parameter N_{ijk}:

$$p(\vartheta_{ij} \mid B_s, \xi) = c \bullet \prod_{k=1}^{r_i} \vartheta_{ijk}^{N'_{ijk}+N_{ijk}-1} \tag{10}$$

Thanks to this algorithm, which is easily implemented by computer programs, the distribution re-adjust its shape according to data provided by available sample data. It exploits the notion of experience, that is quantitative memory which can be based both on quantitative expert judgment and past cases. The prior parameters in the first version of the network are set with a particular equivalent sample size based on expert knowledge (by tuning the values N'_{ijk}). Then subsequent data are added, starting from the given equivalent sample size, so that the two Dirichlet parameters reshape the probability density function according to observations included in the dataset, which determine the values of the set of parameters N_{ijk}. Furthermore, it can be shown that the probability that $X_i=k$ and $\Pi_i=j$ in the next case C_{m+1} to be seen in the database (i.e. expected value) is computed by means of the equation:

$$P(C_{m+1} \mid D,\ B_s,\ \xi)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{N'_{ijk}+N_{ijk}}{N'_{ij}+N_{ij}} \tag{11}$$

where $N'_{ij}=\sum_{k=1}^{r_i} N'_{ijk}$, and $N_{ij}=\sum_{k=1}^{r_i} N_{ijk}$.

4. Practical implementation of the Bayesian predictor

According to what reported in sub-section 4.2, the conditional probability tables of Bayesian Networks (BNs) might be defined from a combination of a priori knowledge about the involved physical processes and a dataset of collected measures. Part of the dataset is usually used for probability estimation, while the remaining part is usually used for validating the BN. Such validation may be done according to the following steps:

1. instantiation of the BN with collected evidences;
2. evidence propagation through the Bayesian reasoning engine;
3. evaluation of estimation/prediction performance by computing a proper performance index.

Once each instantaneous index is computed, a global performance index for the whole validation dataset might be computed for each node.

4.1. The metrics used for validation

The performance of a prediction model may be evaluated by using different kinds of indices. As well known, referring to prediction of a variable at a certain time i, the difference between the predicted value $\hat{X}_i$ and the actual value X_i is defined as error $E_i \triangleq \hat{X}_i - X_i$. Since the validation of a prediction model is interested just in the magnitude of the error, either its absolute value called absolute error $AE_i \triangleq |E_i|$ or its squared error $SE_i \triangleq E_i^2$ can be used. Percentage error is here defined as the ratio (%) between the error and the actual value of the variable: $PE_i \triangleq 100 \cdot E_i / X_i$. In order to have a global performance index to be evaluated over

http://dx.doi.org/10.5772/58470

the whole validation dataset made up of K samples, these instantaneous indices must be combined into global indices. Therefore, the mean absolute error is defined as:

$$MAE \triangleq \frac{1}{K}\sum_1^K |\hat{X}_i - X_i| \tag{12}$$

and the root mean square error is:

$$RMSE \triangleq \sqrt{\frac{1}{K}\sum_1^K (\hat{X}_i - X_i)^2}. \tag{13}$$

The indices of the predicted variables are related to different physical quantities with different units. For this reason, these indices must be normalized with respect to their typical range of variation. Two more indices are defined and practically used for evaluating prediction models:

$$NMAE \triangleq \frac{\frac{1}{K}\sum_1^K |\hat{X}_i - X_i|}{|X_{max} - X_{min}|} \tag{14}$$

$$NRMSE \triangleq \frac{\sqrt{\frac{1}{K}\sum_1^K (\hat{X}_i - X_i)^2}}{|X_{max} - X_{min}|}. \tag{15}$$

ASHRAE Guideline 14-2002 [24] establishes that for calibrated simulations, the *CVRMSE* and *NMBE* of energy models shall be determined for each calibration parameter by comparing simulation-predicted data to the utility data used for calibration. The proposed indices are the coefficients of variation of the root mean square error (*CVRMSE*) and normalized mean bias error (*NMBE*). All these indices are normalized with respect to the arithmetic mean of the variable. Following this guideline, the *RMSE* has been selected as the main performance index for evaluating the accuracy of a BN. However, the mean value of a variable is a good normalization factor only if the variable is always positive (or always negative). For this reason, the range of the considered variable has been taken as a normalization factor and the *NRMSE* has been selected as final index for the design process of the BN: it includes information about both bias and variance of the error.

4.2. Instantiation of BNs for MPC

The control block in the framework depicted in Fig. 1 needs to evaluate a cost function for each candidate control policy. However, since discrete nodes are often used in BNs for describing nonlinearities, the resulting cost function may also present significant nonlinearities. One of the main problems in MPC consists in guaranteeing closed-loop stability for the controlled system. As stated in [25], stability about MPC requires some continuity assumptions. The usual approach to ensure stability is to consider the value function of the MPC cost as a candidate Lyapunov function. Even if closed-loop stability is guaranteed in some way without the continuity assumption, the absence of a continuous Lyapunov function may result in a closed-loop system that has no robustness. This implies that, whenever possible, a continuous cost

function is highly recommended in MPC: it implies better stability and robustness and faster convergence of the optimization algorithm. Moreover, from a modelling point of view, the real buildings are usually continuous systems, or at least hybrid systems (i.e. mixed discrete-continuous) but never pure discrete: a discrete approximation would produce a big variance of the prediction error.

Due to these reasons, the BNs must be continuous with respect to continuous variables. When BNs contain discrete interval nodes (this is often the case in practice, since it allows to model more complex relationships between nodes), an instantiation as "*hard evidence*" (i.e. one state only of a node is assigned 100% likelihood while the other states are given 0%) produces an abrupt variation of the outputs only when passing from one interval to the next one, thus showing a discontinuous behaviour. Therefore, an instantiation must be always done as "*soft evidence*" (i.e. more than one state of a node is instantiated, in fact shaping a probability density function). The idea, with reference to Fig. 5, is to approximate each evidence with nominal value μ and estimated standard deviation σ as a Gaussian distribution $N(\mu;\sigma)$. This distribution is then approximated by an interval discrete distribution that is actually instantiated into the BN as "soft evidence". When the evidence has been propagated, the beliefs retrieved by the network are used to iterate the prediction for an arbitrary number P of steps (see Fig. 7). The final prediction sequence is then summarized as a sequence of normal continuous distributions $N(\mu_i;\sigma_i)$, $i=1, \ldots, P$.

Technically, the key challenge is approximating a continuous probability density function by means of a discrete one, as reported in the following. Given mean value and standard deviation of a normal distribution $N(\bar{\mu}_i;\bar{\sigma}_i)$, and an interval discrete random variable defined by the values $[x_0, x_1, \ldots, x_L]$ (that corresponds to the $L+1$ bounds of the L intervals), the probabilities of each interval must be found such that mean value and standard deviation of the discrete distribution are as close as possible to $\bar{\mu}$ and $\bar{\sigma}$ respectively. As well known, a normal Gaussian probability density function (pdf) takes the form $f(x)=1/\sqrt{2\pi}e^{-x^2/2}$, $-\infty<x<\infty$.

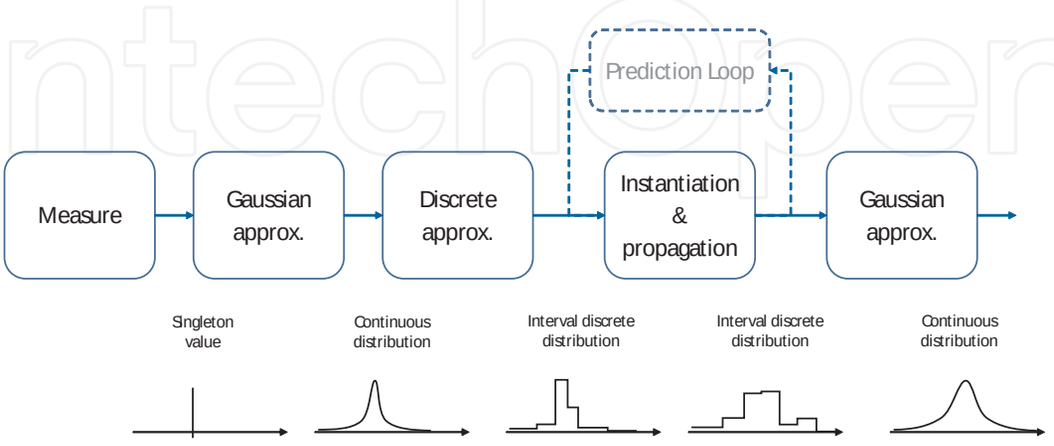

Figure 5. Instantiation of a measure as "soft evidence" and corresponding information flow. Instantiation and propagation is repeated in the prediction loop (dashed), as described in Fig. 7.

http://dx.doi.org/10.5772/58470

The cumulative distribution function (cdf) is then given by $\Phi(x)=1/\sqrt{2\pi}\int_{-\infty}^{x}e^{-t^2/2}dt$ that has no close form. An effective and sufficiently accurate way for approximating it is provided by [26]:

$$\Phi'(x)\approx\frac{e^{2z'}}{1+e^{2z'}},\ \text{where } z'=0.7988x\left(1+0.04417x^2\right) \tag{16}$$

In order to consider a general distribution with mean μ and standard deviation σ, the variable x must be replaced with $\frac{x-\mu}{\sigma}$:

$$\Phi(x)\approx\frac{e^{2z}}{1+e^{2z}},\ \text{where } z=0.7988\left(\frac{x-\mu}{\sigma}\right)\left(1+0.04417\left(\frac{x-\mu}{\sigma}\right)^2\right). \tag{17}$$

Given any standard deviation $\bar{\sigma}>0$, and mean value $\bar{\mu}>0$ to be approximated, the value of standard deviation must be at least greater than the standard deviation σ_{min} of the interval $[x_l;x_{l+1}]$ in which $\bar{\mu}$ falls:

$$\sigma=\max\left(\bar{\sigma},\ \sigma_{min}\right),\ \sigma_{min}=\frac{x_{l+1}-x_l}{\sqrt{12}}. \tag{18}$$

For each interval of a standard discrete node, probability is given by $q_i=\frac{q_i'}{\sum_{i=0}^{N-1}q_i'}$ where:

$$q_i'=\Phi(x_{i+1})-\Phi(x_i) \tag{19}$$

For a discrete periodic variable (like a direction) defined in the interval $[x_0;x_L]$ (that for a direction is $[0;2\pi]$), the pdf can be split into periodic intervals shifted each other by integer multiples of period (x_L-x_0):

$$[x_0+k*(x_L-x_0);x_L+k*(x_L-x_0)],\ k\in Z \tag{20}$$

If standard deviation is sufficiently smaller than the period $\sigma\ll(x_L-x_0)$, this infinite series can be approximated by the three central intervals, that is for $k=-1,0,1$. Moreover, when $\sigma\cong(x_L-x_0)$, probability is almost equally distributed all over the interval and the approximation introduced by neglecting the stubs for $k<-1$, $k>1$ can be simply compensated by renormalizing the resulting distribution. Thus, in this case, probability of each interval is given as in the previous algorithm but the right and left stubs of the Gaussian are overlapped in order to consider periodicity of the discrete variable. For each interval of a periodic discrete node, probability is given by $p_i=\frac{p_i'}{\sum_{i=0}^{L-1}p_i'}$ where:

$$p_i^{'} = (\Phi(x_{i+1}) - \Phi(x_i)) + (\Phi(x_{i+1} + (x_L - x_0)) - \Phi(x_i + (x_L - x_0))) + (\Phi(x_{i+1} - (x_L - x_0)) - \Phi(x_i - (x_L - x_0))) \quad (21)$$

The instantiation strategy described above was used according to the scheme in Fig. 5 for implementing the discrete approximation and smoothing the behaviour of the BNs when inputting evidences. The first Gaussian approximation of Fig. 5 has mean value equal to the measure (eventually calibrated) and standard deviation represented by the uncertainty of the measurement process, that can be determined by the data-sheet of the installed sensor. The second Gaussian approximation is trivially implemented by computing mean and standard deviation of the input distribution. Fig. 6 shows examples of Gaussian distributions approximated so as to be assigned as evidences of a Bayesian Network.

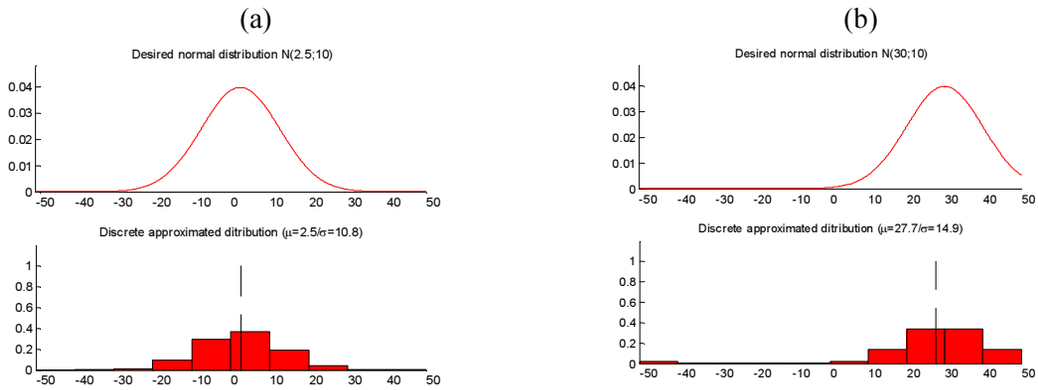

Figure 6. Comparison between the desired Gaussian distribution and the approximated one to be instantiated in an interval discrete chance node of a BN: example of a non-periodic variable (a) and of a periodic variable (b).

5. Development of the Bayesian models and cost function

5.1. The procedure

Basically, the development of both Dynamic Bayesian Networks (DBNs) and regular BNs is not an easy task and usually consists of three main phases:

1. definition of the network topology (structural learning);
2. preparation of the training set and learning of the conditional probability tables;
3. final assessment of the network.

In the case object of this chapter, the behaviour of the metro station Passeig de Gracia was first simulated through whole building analyses, which provided datasets encompassing all the possible environmental conditions, including those which are considered as not very likely, then that knowledge was transferred into Bayesian Networks. This was deemed necessary

http://dx.doi.org/10.5772/58470

because for such complex domains eliciting expert knowledge to learn conditional probability tables is not feasible. Hence several sets of data were generated through simulations prior to the application of the EM learning process whose main features were presented in sub-section 4.2. Three datasets were generated:

- the first one was made up of randomly generated data, which means that the inputs (e.g. weather, heat gains, occupancy figures etc..) were allowed to vary without additional constraints within their range;
- the second sample, called "likely" dataset, was generated through simulations whose inputs were allowed to vary within their same ranges cited above, but their differential variations being constrained: it means that the difference between the value of each variable at the present time step and the value of the same variable at the previous time step was limited by a threshold;
- the third "typical" sample was built through simulations, whose inputs were taken from real measurements, such as real weather conditions in Barcelona, number of people passing through the station and trains recorded by occupancy surveys, internal heat gains estimated on the basis of real measurements etc..

As the learning process got information from all these samples, the resulting networks not only were expected to be able to respond to sudden variations of external actions and to consider quite unlikely events (which is the case of the first dataset) but to be very sensitive to the more likely scenarios included in the second and third datasets.

The whole building model used for running simulations was developed within the Seam4us research as a lumped parameter model in the Dymola™ simulation environment, that is based on the Modelica language [27]. Starting from a validated library for building simulation developed by the Lawrence Berkeley National Laboratory [28], a specific library for underground station was developed, which is so far in its validation phase.

However, such a model cannot be run in real-time when the controller needs to determine the best candidate control strategies, so it was reduced into the less computationally demanding form of Bayesian Networks. In order for the PdG-L3 model to predict the environmental status of the station, it was split into two Bayesian Networks, each relative to a different physical phenomenon:

- temperature prediction dynamic Network (TP-DBN), which is in the form of a DBN, because it forecasts expected temperature in the station given inputs about current and past time steps;
- air flow prediction Bayesian network (AF-BN), which is in the form of a regular BN, because it estimates variables relative to air flow in the station and energy consumption of the fans, given its current status.

Once available, the two networks were run according to the scheme outlined in Fig. 7. Here a portion of the whole process in Fig. 1 is depicted. At every iteration the controller will opportunely query the two of the networks to get future estimations about the variables

relevant to select the most opportune control policy to be adopted at each running step. To this aim, the networks need to be instantiated first: the current temperature in the station's platform (PL3) and weather conditions will be provided by the permanent monitoring network installed in the station, along with candidate fan frequencies. Given these inputs, the controller is allowed to query the AF-BN in order to estimate fans consumption and air changes in the station at each time step. Such a prediction step takes a few seconds and is performed by the software Hugin™ through algorithms for belief propagation, of the kind already reported in sub-section 4.1. Then, the TP-DBN will take these variables as inputs, along with other state variables (e.g. current PL3 temperature, temperature difference between inside and outside and forecasted weather, people, train arrival etc..) in order to predict PL3 temperature at the next time step (which is typically every hour, unless a different control step is required). Again belief propagation is performed with this second network. Then, the same loop will be repeated at each iteration. This loop is based on the assumption that temperature variations between two consecutive hours is so small to be considered constant by the AF-BN without big prejudice. In other words, to the AF-BN purposes temperature at the next time step (T1) was considered equal to temperature at the current time step (T0). This assumption did not compromise the reliability of future estimations, as will be shown by the validation at the end of this sub-section. Both the networks were built following the same methodology:

1. first structural learning: it was determined first by the a-priori knowledge from the researchers, who ordered and connected the variables according to the physical relationships among them; then statistical analyses were used to improve the assumed structure and discern the strongest relationships among the variables themselves;
2. improvement of the network's structure: this was carried out through analysis of its performance indices, after learning conditional probability tables from the random dataset, which was the first set of data generated by running the Dymola™ model without constraints on all the possible weather conditions and disturbance actions;
3. final refinement using the two additional datasets: adding more datasets allowed the developers to quantify even probabilistic relationships among the variables and to define the final shape of the networks;
4. final evaluation of the networks: as for the three steps above, also in this final step the indices defined in paragraph 5.1 were used to evaluate the prediction capabilities of the networks.

The first step started from the analysis of 81 variables included in the Dymola™ dataset, properly filtered and re-sampled at a 60 sec rate. Iterative cluster analyses [29] were useful to group those variables into clusters and determine those which were redundant [30], because providing the same information given by others (e.g. surface temperatures in PL3 were highly correlated and grouped in the same level, hence just one of them was kept and the remaining ones were dropped out).

This first step allowed us to cut down the number of involved variables to 39. Then, a further reduction was done, where only those variables showing the strongest dependence relationships were kept, cutting the whole number down to 25, hence finding the minimum number

http://dx.doi.org/10.5772/58470

of nodes required to be included in the Bayesian graphs. This final set was naturally grouped into two sub-clusters of variables: one of them including those related to air flow processes, and the other one including those related to the environmental temperature dynamics. Finally, the qualitative relationships among variables (represented by arcs in the networks) were worked out partly by expert knowledge, and partly using the "structural learning" tool supplied by Hugin™, which used the information included in the "random" dataset.

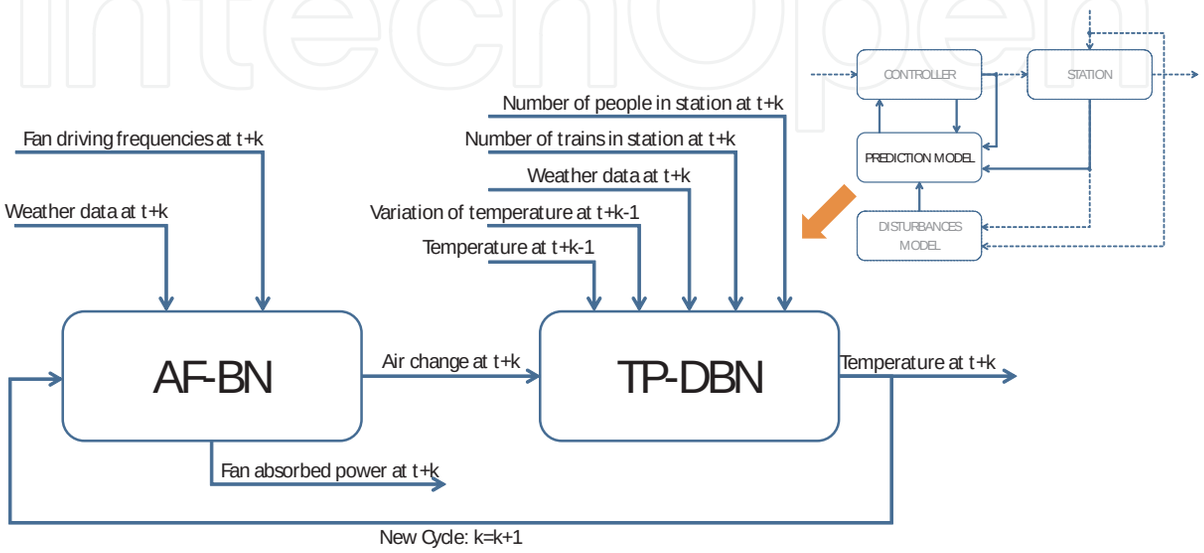

Figure 7. The basic loop of the predictive cycle adopted for PdG-L3 which involves both Bayesian Networks. It implements the prediction model in Fig.1, which starts by setting a counter at 1 and loops until the counter reaches any desired prediction horizon.

The second step was the most critical in the development process, and helped in several tasks:

- meaning and dependencies between nodes have been reviewed according to the relationships suggested by physical laws (e.g. PL3 temperature is affected by the amount of outside air supplied by ventilation and by the temperature difference between indoors and outdoors, which became its parents);
- the number of intervals for discretizing the state space pertaining to every node, which was refined through an iterative process, with the final purpose of minimizing the errors of the output variables given by the indices outlined in sub-section 5.1;
- a few links have been rearranged, in order to improve the performances of the two networks.

Fig. 8-a depicts the final structure of the dynamic Bayesian network (i.e. TP-DBN), which was used to predict PL3's temperature in PdG station in the next step (node TemPL3_p01), starting from inputs such as: forecasted number of people in the station at the next step (NPeSta_p01), forecasted internal gains supplied by trains at the next step (GaiTr_p01), current PL3's temperature (TemPL3), forecasted outdoor temperature (TOuMet_p01), forecasted air changes per hour (ACOPL3_p01) and deviation of temperature from the past time step (DTePL3). The

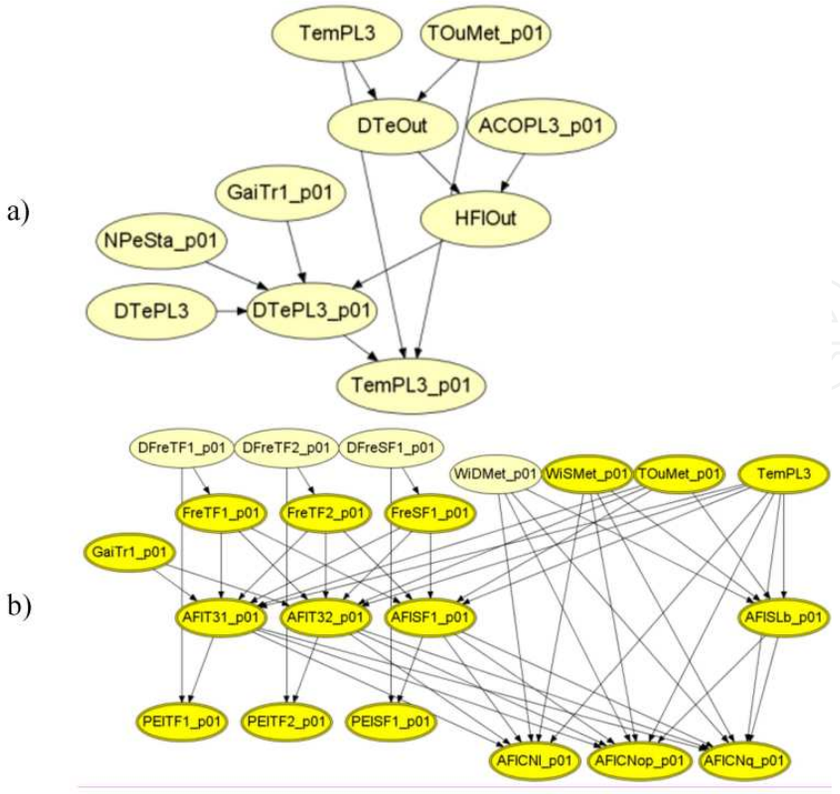

Figure 8. Predictive and dynamic Bayesian Network relative to temperature in PL3 (a) and predictive Bayesian Network relative to air flow changes in the station (b).

network's intermediate variables are useful to perform computations and simplify conditional probabilistic relationships among variables.

Similarly holds with the AF-BN network (Fig. 8-b), whose inputs are: forecasted frequencies of fans in the station and tunnels at the next time step (DFreTF1_p01, DFreTF2_p01, DFreSF1_p01), forecasted internal gains by trains (GaiTr1_p01), forecasted wind direction and speed (WiDMet_p01, WiSMet_p01), outdoor temperature (TOuMet_p01) and current temperature (TemPL3). The main outputs are the power consumption of fans – in the station (PElSF1) and in the tunnels (PElTF1, PElTF2)-and air flow rates expected across the corridors leading to PL3: AFlCNl_p01 (corridor CNl), AFlCNop_p01 (sum of corridors CNo and CNp), AFlCNq_p01 (corridoio CNq) and AFlSlb_p01 (station link). These estimated airflows are then summed up coherently to the Air Mass Balance for computing the overall air change in PL3 (ACOPL3), needed as input from the TP-DBN (Fig. 7).

During this process, whenever expert knowledge deemed more than one structure as reasonable for a certain network or fragment of it, the best one was chosen by evaluating which of them returned the highest performances according to the indices described in sub-section 5.1, after learning from the random datasets, which is the one where the widest variations and fastest dynamics of variables were considered. Even the other aforementioned amendments (e.g. rearrangement of links, optimal discretization of the networks' nodes etc..) were tested through optimization of the performances indices defined in sub-section 5.1.

http://dx.doi.org/10.5772/58470

The third step was aimed at performing further refinement using the "typical" and "likely" datasets, which were generated through Dymola™. Technically, that means that the EM learning algorithm was performed by adding the information included in these two datasets to the information already derived from the "random" dataset. This process allowed to include more information in the networks, mainly about those scenarios which are likely to occur more often. So it helped make the model more accurate to predict those states which are more likely to occur.

The refinement was mainly performed in terms of tuning the subdivision into intervals of the nodes and in terms of converting discrete variable into continuous variables, should they work better. The steps no. 2 and 3 required many iterations of learning, refining and validating.

Each loop was characterized by either a modification of the structure or of any network's node and conditional dependencies. Then the modified network's version was evaluated to define whether the modification had to be kept (in case it decreased the errors) or had to be rejected (in case it increased the errors), being the errors estimated as explained in section 5.1.

On the whole and just to give an example, 140 cycles were made with the TP-DBN (Tab. 1), which was useful to reduce the error from 4.98 ° to 0.72 °C (in the *RMSE* case) and from 17% to 4% (in the *NRMSE* case). The trend during the refinement process led to a continuous increment of performances, as shown in Tab. 1. In addition, 82 cycles were needed to optimize the AF-BN: for the control variable, Station Fan Power (PElSF1) *RMSE* fell down from 1858 W to 377 W, whereas *NRMSE* fell from 10.3% down to 2.3%.

In the TP-DBN all the nodes were represented by discrete variables. In the AF-BN all the variables were continuous except the following ones: frequencies of fans (DFreTF1_p01, DFreTF2_p01, DFreSF1_p01) and wind direction (WiDMet_p01).

Cycle no.	TemPL3_p01	
	RMSE (°C)	NRMSE (%)
1	4.98	17
…		
54	3.32	12
…		
98	1.81	6
…		
114	1.00	4
…		
140	0.72	4

Table 1. Gradual improvement of performances during continuous refinement of the TP-DBN network.

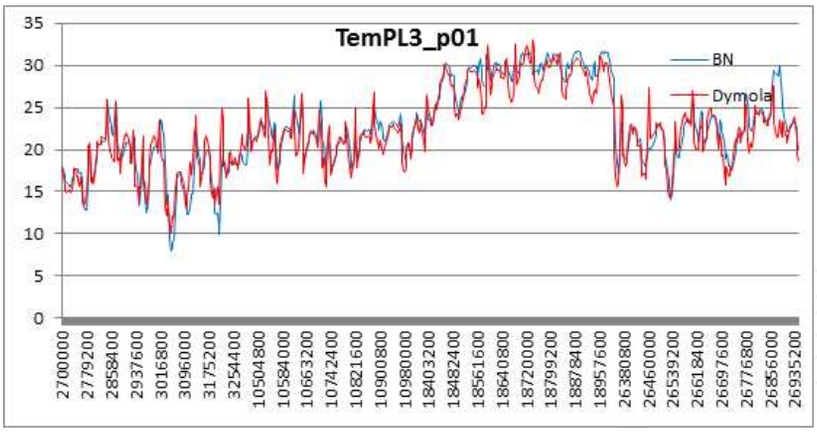

Figure 9. Qualitative comparison between the real temperature plot computed by Dymola™ and forecasts by the Bayesian Network TP-DBN.

Finally, and as fourth step, the performances of the two networks were verified also through simulations. Fig. 9 shows the good agreement between the real temperature simulated by Dymola™ in PL3 and the forecasted plot of PL3 as predicted by TP-DBN. The simulations performed by the Bayesian Networks in this case were carried out according to what already described. The input values at the first time step were instantiated as evidences taken by the Dymola model. Then, the outputs from the networks were used as inputs for the next time step in the networks and the simulations were iterated in the same way all over the period shown in the diagram. It's clear that the predictive and dynamic Bayesian network are able to accurately model the temperature plot in PL3 and to give the right inputs to the controller, in order to evaluate the best control policy.

5.2. The cost function

With reference to Fig. 1, the controller unit passes a candidate control policy to the BNs and uses resulting predictions in order to compute a cost function, which must select the best output to be used as an input in the next time step. The degrees of freedom (outputs) of the controller for PdG-L3 station are the frequencies of the station fans (*FreSF*1, *FreSF*2). The predictions that the controller queries to the Bayesian Networks are the absorbed powers of tunnel fans and station fans (*PElTF*1, *PElTF*2, *PElSF*1, *PElSF*2) and the air temperature in the platform (*TemPl*3). The future outdoor temperature (*TOuWS*) is retrieved from a weather forecast service and the air change in the platform (*ACOPl*3=amount of clean air entering the platform) is computed as a proper combination of the air flows predicted by the BNs (derived by the specific station topology):

$$\begin{aligned} ACOPl3 \triangleq\ & AFlSFa1^{+} + AFlSFa2^{+} + AFlCN1^{+} + \\ & +[(AFlCNop - AFlSLb + AFlCNq)^{+} - \mid AFlSL\ b^{-} \mid]^{+} \end{aligned} \tag{22}$$

The objective of MPC is to minimize the following cost function with respect to station fan frequencies:

http://dx.doi.org/10.5772/58470

$$J = \sum_{k=1}^{H} \alpha_{PT}\left(\frac{|PElTF1(k) + PElTF2(k)|}{\tilde{}_{2PT}}\right) + \alpha_{PS}\left(\frac{|PElSF1(k) + PElSF2(k)|}{\tilde{}_{2PS}}\right) +$$
$$+\alpha_{DT}\left(\frac{TOuWS(k) - TemPl3(k)}{\tilde{}_{DT}}\right)^2 + \alpha_{T}\left(\frac{\bar{TemPl3} - TemPl3(k)}{\tilde{}_{T}}\right)^2 + \tag{23}$$
$$+\alpha_{AC}\left(\frac{\bar{ACOPl3} - ACOPl3(k)}{\tilde{}_{AC}}\right)^2 + \alpha_{DF}\left(\frac{FreSF1(k) - FreSF1(k-1)}{\tilde{}_{DF}}\right)^2$$

Subject to constraints:

$ACOPl3(k) > ACOPl3_{Min}$

$TemPl3(k) < TemPl3_{Max}$

$FreSF_{Min} < FreSF1(k) = FreSF2(k) < FreSF_{Max}$

The variables marked with "tilde" (~) are the normalisation coefficients that corresponds to the typical values of the corresponding variable. At sampling time $t \in \mathbb{N}$, $PElTF1(k)$ represents the estimation of the value of variable $PElTF1$ at time $t+k$ ($k \in \mathbb{N}$) evaluated at time t (i.e. $PElTF1(t+k \mid t)$). The design parameters of the MPC controller are the prediction horizon H, the desired values denoted with bar notation $\bar{TemPl3}$, $\bar{ACOPl3}$, the weights of each single objective in the cost function $\alpha_{...}$, and the bounds of the given constraints: $ACOPl3_{Min}$, $TemPl3_{Max}$, $FreSF_{Min}$, $FreSF_{Max}$.

5.3. The networks and their estimates

The "*soft evidence*" instantiation strategy presented in section 5.2 has been implemented in a java library that wraps Hugin™ reasoning engine and allows also for other high-level functionalities, like multiple network iterations and interconnection between different networks that shares the same variables. This library is able to get the set of variables describing the current state of the station and to initialize with them the first network to be queried. Then, it is able to perform probabilistic inference by running the Hugin™ application and to extract the outputs, which will be transferred to the second network to be queried according to the procedure already shown in Fig. 7. This is done H times, if H is the desired prediction horizon.

The same functions has been also integrated in an excel spreadsheet for validating BNs. The two networks were combined according to the scheme depicted in Fig. 7 and used to simulate the predictive control. Fig. 10 shows the prediction results for one typical day (in blue), i.e. prediction horizon H=24 hours, compared with the simulation results achievable from the Dymola model (shown in red), regarding the main outputs: expected energy consumption by one of the station' fans (6a), overall airflows coming from outdoor air (6b) and future temperature (6c) plots in the platform of Line 3. In order to assess the level of accuracy of such predictions, Table 2 shows the corresponding *RMSE* and *NRMSE* relative to the variables

plotted on Fig. 10. The agreement between the real plots and the estimated one is very good at each prediction time, especially for the energy consumption, that is one of the key variables influencing the cost function described in paragraph 6.2 and, as a consequence, the results of the loop managed by the controller.

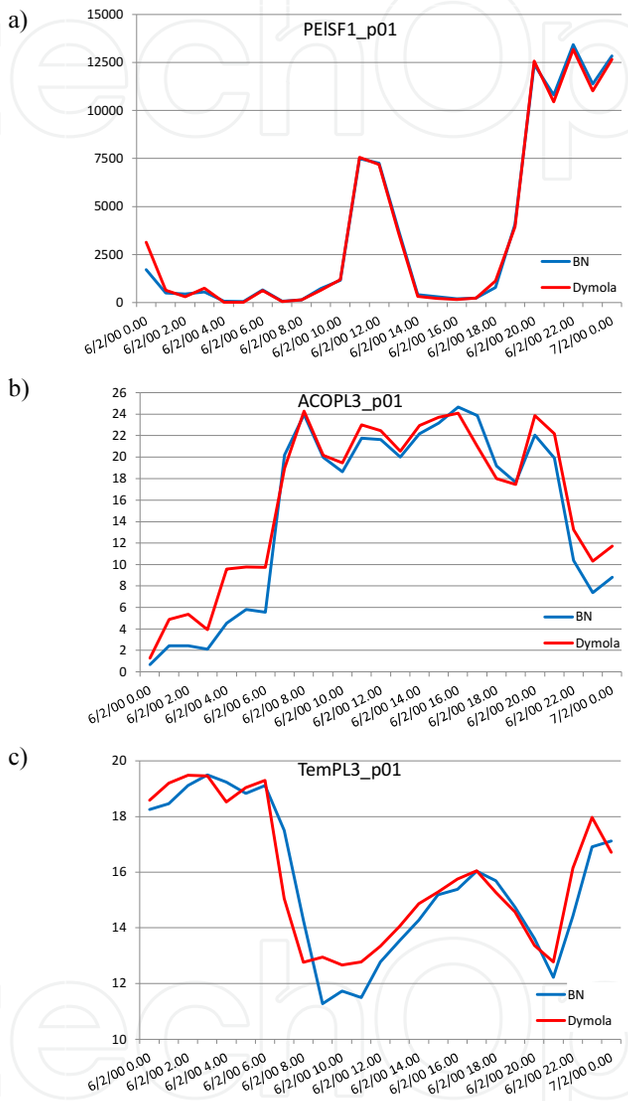

Figure 10. Comparison between real energy consumption of station fan PELSF1 and the estimated one by AF-BN (a), the real air change per hour and the estimation by AF-BN (b) and comparison between the real PL3 temperature and the estimation by TP-DBN (c).

PElSF1_p01		*ACOPL3_p01*		*TemPL3_p01*	
RMSE [W]	*NRMSE* (%)	*RMSE* [kg/s]	*NRMSE* (%)	*RMSE* [°C]	*NRMSE* (%)
326.28	0.024784	2.24	0.097566	0.91	0.13406

Table 2. Errors obtained for the whole prediction cycle with H=24 hours in the three prediction cases shown in Fig. 10.

http://dx.doi.org/10.5772/58470

6. The Passeig de Gracia simulator case study

In this section, we will report the implementation of a Bayesian Network predictor for the environmental control of the aforementioned test-bed given by *Passeig De Gracia* (PdG) metro station in Barcelona, Spain. The purpose of this section is to provide readers with an example regarding how Bayesian Networks can be embedded within a large MPC control framework, and how their predictions can be exploited for optimal control. The environmental control has been based on MPC because of the great complexity of the thermal and airflow dynamics in the underground environment, and of the time length of their characterising constants. The metro station underground environment is mainly characterised by huge thermal inertia caused by the terrain surrounding the station, and by relevant contributions to the indoor temperature and pollutants levels provided by air flows. Indoor air flows are determined by a number of sources that influence the station with different time frequencies and during different daytimes. The outdoor wind flow, quite frequent in a coastal station like Barcelona, affects directly the shallowest levels of the station, and determine the air pressure configuration at the interfaces of the deepest levels that may favour or impede the mechanically air supply, depending on their mutual disposal. The train transit causes the well-known piston effect. The approaching trains act as pistons compressing the air into the station platform, and leaving trains act as pulling pistons, reversing the flows. The frequency of this effect depends on the train schedule and, in our case, can be assumed about 180 sec on the average. The train piston effect causes relatively high-speed air transients with many local turbulences that affect also the platform neighbouring spaces. A third relevant source of air exchange, which is usually neglected in standard analyses, is due to the air buoyancy caused by the temperature difference between the indoor and the outdoor environments. This effect becomes substantial during the night, because of the absence of the other sources and of the greater temperature difference. Buoyancy effect causes relatively slow and laminar airflows that move air form the rail tunnels to the outside through the station's spaces.

Within this scenario, the problem of controlling the forced ventilation in an optimal way, through the fan speed, cannot be approached but with an MPC technology. Within the MPC framework, the definition of the weighting coefficients of the cost function, that maximize energy saving without compromising comfort and air quality in different operating conditions (see section 6.2), and without affecting normal operation of the station (due to safety reasons), required the development of a co-simulation architecture. Thus, the Bayesian predictor and the MPC logics has been embedded in a simulation environment that accurately reproduces the thermal and air-flow dynamics of the outdoor and indoor environments, and the trains and passenger flows. The development of the models that contribute to the simulation environment required in depth preliminary analysis by means of Finite Element Modelling, and a number of on-site surveys, that became necessary to determine the magnitude of the phenomena and to subsequently calibrate the models. Once they have been calibrated and included in the simulation architecture, the environmental models resemble the same dynamic of the measured environment, thus allowing for scenario analysis and control sizing.

6.1. The SEAM4US simulator

The Simulink (Mathworks©) architecture of the SEAM4US simulator is shown in Fig. 11. The simulator is made of four main components. The PdG Environmental model, the passenger flow simulator, the lighting control simulator and the environmental MPC.

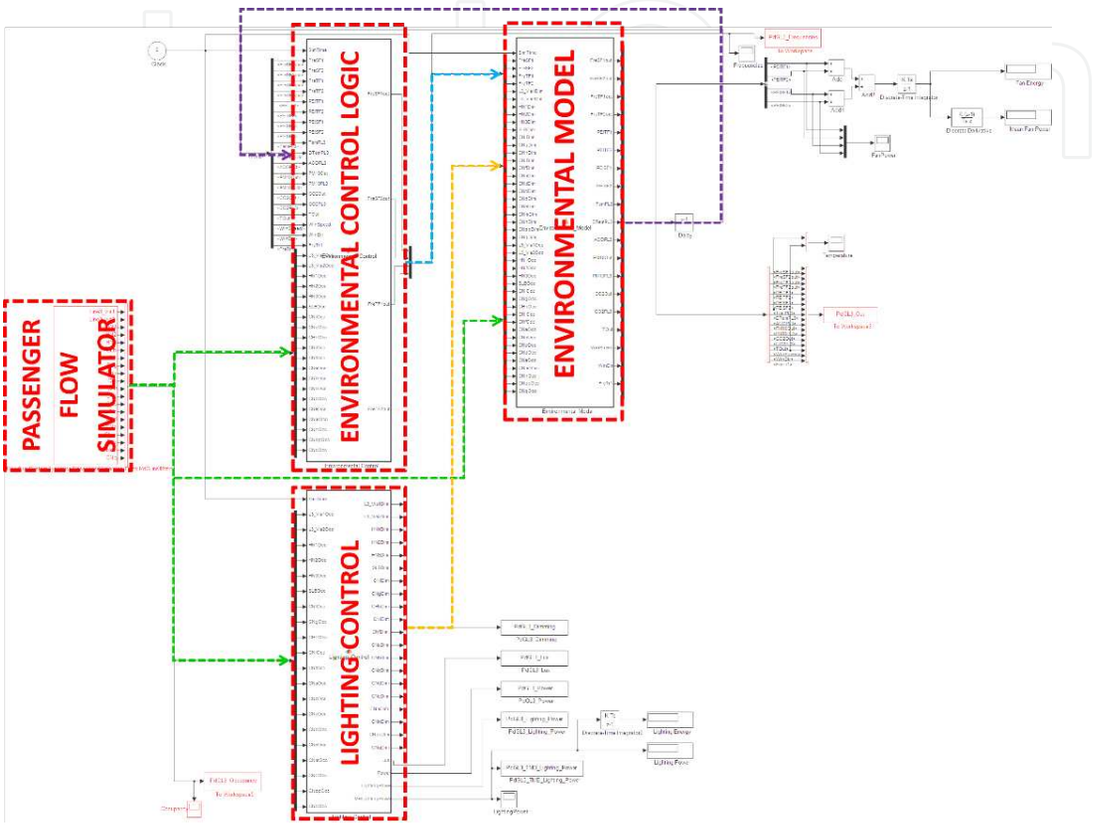

Figure 11. The Simulink SEAM4US Simulator architecture: occupancy (green), fan frequencies (blue), dimming level of lights (orange), measures (violet).

The PdG Model is the one developed in the Modelica language. To this aim, the Buildings Library 1.3 [28] has been extended in order to represent physical entities and parameters of the underground environments. At compile time the PdG environmental model results in a matrix with tenths of thousands of unknowns. The PdG Model is interfaced with a weather file of Barcelona that provides the hourly external weather parameters, including wind speed and directions. The PdG Environmental Model receives as inputs the passenger occupancy levels of each space of the station, the lighting level of the appliances in each space, and the fan control frequencies. It then outputs all the environmental parameters, like air temperature and humidity, the pollutants levels, and the energy consumption of the fans. These parameters are then fed back to the control logics as the basis for the next control step. In the SEAM4US simulator the large PdG Environmental model acts as the real station. In principle, this model could be used in the deployed MPC system to provide the necessary and accurate predictions of the next future in the model predictive controller. This option has revealed unpractical. The big size of the model produces a large memory footprint and requires significant computation

http://dx.doi.org/10.5772/58470

power to get reasonable and effective simulation times. This is in contrast with the model embedding requirements, which foresees models included in relatively light computational environment due to deployment constraints and cost reasons. Furthermore, the alignment of the initial state of such a large model with the actual state of the station is very problematic in terms of computational time and of the stability of the solution. Therefore, the model that support the controller by means of predictions on the future status of the station has been derived from the PdG environmental model through a model reduction process into Bayesian Networks, as described in Section 6 of this Chapter.

The PdG has been used to produce a large set of control cases that have been analysed to find out the minimum set of parameters for an effective control of the target performances. The reduced case set has been then fed into the Bayesian Network, through EM learning algorithms, in order to get the Bayesian predictor for the MPC control shown in sub-section 5.1. The size of this predictor is small enough and its computational time short enough to suit the model embedding requirements. The statistical nature of the predictor avoids any problems concerning the estimation of the initial state.

The prediction accuracy achieved by the reduced model is good enough to get a reliable control of the station.

The passenger model simulates the passenger flows and the consequent occupancy distribution of the spaces inside the station. It is regulated by the train schedule, by the hour of the day – either a normal or a rush hour-and by the week day-either a weekend or a working day. The simulator has been developed in the Modelica language and is based on the bond graph theory. The passenger flow is simulated as a mass flow occurring among the station spaces. The mass sources are modulated by train arrivals and hourly scheduled flow rates observed from the outside. The model calibration has been carried out through observations (i.e. sampling performed by means of security cameras installed in the metro station) of the flow rates of passenger entering and exiting the trains and the station entrances at different hours of the days. The internal flow is then regulated through mass flow delays calculated on the basis of typical transit speeds.

Finally, the lighting control subsystems regulates the lighting level adaptively in relation to the occupancy level of each station ambient. It implements a reactive form of control that is driven by the visual task of each specific situation that may occur in the station ambient, either rush hour, waiting train arrival, and so on. The outcome of the lighting simulator are the dimming level of each appliance of the station. This of course influences the station environment as a thermal gain and is therefore provided to the PdG environmental model input.

A noteworthy aspect of the SEAM4US simulator is its computational arrangement. In fact, it implements a co-simulation architecture. The overall container is the Simulink system, which provides the control clock and a fixed simulation step to all the other subsystems. The passenger model and the lighting simulator are included in the Simulink framework as Functional Mock-up Units (FMU) that internally include the solver, through Functional Mock-up Interfaces (FMI). Interestingly, the FMI protocol provides the necessary interface between the different solvers, adapting the time varying simulation steps of the FMUs with the Simulink

fixed one. The inclusion of the large PdG environmental model was not that easy. The Dymola model did not allow exporting the solver that is uniquely able to simulate the model within an FMU. Therefore, we have opted for a loosely coupled simulating environment, which runs the Dymola and the Simulink in parallel and synchronize them through a DDE channel. A wrapper of the Dymola model, which essentially implements an FMI, provides the necessary sync and message transfer functionalities.

6.2. How the MPC controller works

The control logics implemented in the SEAM4US simulator is based on a simple particle filtering mechanism (Fig. 12). The controller randomly generates a number of different control options that are sent to the predictor. The predictor updates the model with the control parameters and by means of Bayesian inference calculates the environment and energy consumption parameters. Then the controller ranks the predictor outcomes according to a cost function and to constraint satisfaction. The best performer is then selected and used in the next control step. In the PdG case described in section 6.1, control options are all the driving frequencies of the fans installed in the station and described by variables FreSF1 and FreSF2, as explained in section 5.2. The model predictor is made of the two Bayesian Networks AF-BN and TP-BN described in section 5.1. and depicted in Figs. 7 and 8. Once a set of outcomes (e.g. expected energy consumption, comfort parameters etc.) is had per each value of the inputs, the cost function in eq. (23) selects the best control strategy and sends it to the actuation system. This value is kept constant over one hour, after which another MPC step is run by the controller to re-adjust actuators for the next hour.

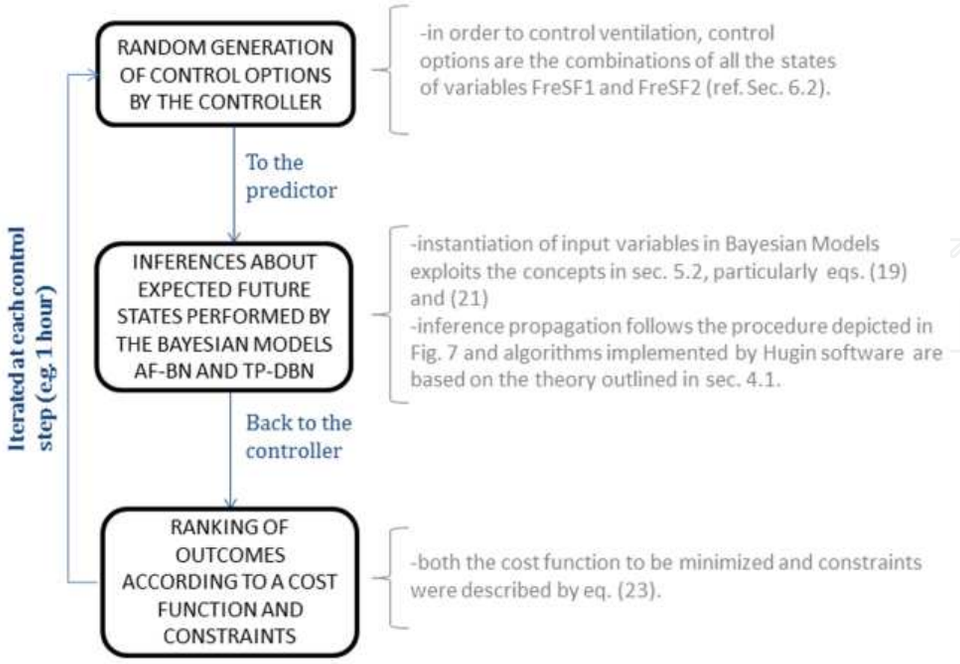

Figure 12. Flowchart showing the procedure adopted by the MPC controller for the PdG case study.

http://dx.doi.org/10.5772/58470

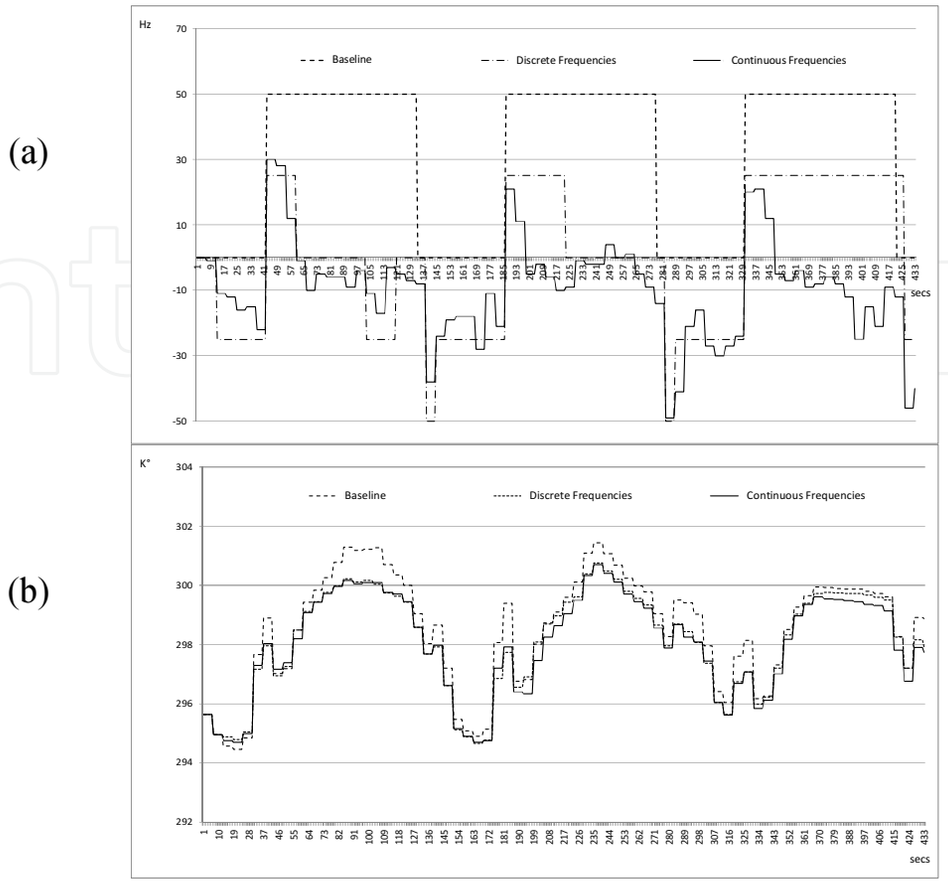

Figure 13. Some plots from the simulations in search of optimal control strategies: frequency of the fans in charge of mechanical air supply according to the control policy (top) and related air temperature plots determined by the most meaningful control strategies (bottom).

Fig. 13 shows an example of a simulation result of three days of operation, which is relative to the environmental control. The simulation time is represented along the x axis, while the y axis represents the fan frequencies in Fig. 13-a. Negative frequencies means that the fan direction is inverted (extracting air instead of supplying). Three curves are reported. The dashed curve (Baseline) depicts the current policy used for fan control, as it is actually implemented in the station, which we assumed as the baseline. The fan is driven at maximum speed for all the station opening time and it is turned off during the closure time. The second dash-dot curve represents MPC constrained to only two driving frequencies, while the third (continuous) curve is related to a continuous frequency driving. In addition to the fact that MPC control provides an energy saving rate that can rise up to 35%, it is noteworthy to realize why this happens. Comparing the baseline curve with the MPC controlled, it appears that in many cases the driving frequencies and the baseline have opposite signs. This means that in the standard baseline driving the station fans very often are opposed to the air flow induced by the external sources, and therefore contribute negatively to the air exchange and to the comfort parameters. This is reflected by the temperature curves that are slightly lower – i.e. more comfortable-for the MPC controlled environment despite the huge energy saving (Fig. 13-b). Summarizing,

these results show how the effectiveness of the MPC control of complex environment relies on the power and on the flexibility of the Bayesian predictor and of the Bayesian Inference paradigm.

7. Conclusions

In this Chapter we have shown that the role of Bayesian predictors may be critical in order to implement predictive control of buildings. This kind of control is one of the most effective ones currently being developed by researchers, because it is able to smooth control actions and to trigger them in advance. However, it cannot be applied without a reliable predictor of the expected state of the controlled domain. Such a predictor must be queried in real-time, which is feasible just in case a reasonable computational effort is required.

In other words, computationally demanding software programs cannot be used to produce predictions at run time, but they can be run to generate datasets and these datasets may be used to transfer knowledge into Bayesian Networks. At this juncture, Bayesian inference may be performed: in fact, inputs by the controller (i.e. input variables describing the current state of the domain plus candidate arrays of control values) are instantiated in Bayesian Networks in the form of a set of evidences; then, inference algorithms are propagated and expected future values describing the energy and thermal state of the domain might be estimated. This procedure can be repeated thousands of times at each control step and it makes the implementation of MPC feasible.

When implemented in a real case, the results from inferences were shown to be very accurate with low deviations from the values estimated by means of more complex numerical models. In addition, our testing of the use predictive Bayesian Networks embedded in a wider MPC framework to support the ranking of concurrent control policies was successful, too. So Bayesian Networks proved to be able to solve the problem of reducing complex models into more manageable tools for performing cumbersome inferences through limited computational efforts, while getting highly accurate results.

8. List of notations

Nomenclature	Meaning
General	
BN	Bayesian Network
DBN	Dynamic Bayesian Network
TP-DBN	Temperature Prediction Dynamic Bayesian Network
AF-BN	Air Flow Prediction Bayesian Network

MPC	Model based Predictive Control
PdG-L3	Passeig de Gracia – Line 3 station
BEMS	Building Energy Management System
HVAC	Heat, Ventilation, Air conditioning and Cooling
Section 2	
y, $\hat{y}$	Outputs of the controlled building and their predictions respectively
u, $\hat{u}$	Inputs of the controlled building and their predictions respectively
d, $\hat{d}$	Disturbances and their predictions respectively
m	Measures from building's sensor network
Section 3	
$P(x)$	Prior probability
X, Y, Z	Random events or prepositions
$M_{y\|x}$	Conditional probability matrix
$P(y_i \mid x_i)$	Any entry of the conditional probability matrix
ξ	Evidence observed before estimating future values of random variables
$BEL\ (x)$	Overall belief accorded to proposition X=x by all evidence so far received by ξ
$\lambda(x)$	Likelihood vector
β	$1/P(\xi)$
U	Domain to be modelled
B_s	Bayesian Network structure ordering a set of variables in domain U
n	Number of variables in B_s
D	Random sample
C_i	Any record of a random sample
Π_i	Any set of variables in B_s which are parents of X_i
q_i	Number of states of Π_i
r_i	Number of states of x
$_{ijk}$	Physical probability that any variable X_i in a B_s will come out with one of its states k, given its joint set of parents are in state j
c	Normalization constant in a gamma function
N	Multinomial parameter in a gamma function
Sub-section 4.1	
$\hat{X}_i$	Predicted value of any random variable X_i
X_{max}	Maximum value of variable X_i

X_{min}	Minimum value of variable X_i
K	Number of samples in a validation data set
E_i	Prediction error at time i
AE_i	Absolute error at time i
SE_i	Squared error at time i
PE_i	Percentage error at time i
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
NMAE	Normalized Mean Absolute Error
NRMSE	Normalized Root Mean Square Error
CVRMSE	Coefficient of Variation of the Root Mean Square Error
NMBE	Normalized Mean Bias Error
Sub-section 4.2	
$N(\mu;\sigma)$	Normal distribution with mean μ and standard deviation σ
$N\left(\bar{\mu};\bar{\sigma}\right)$	Desired distribution with mean value $\bar{\mu}$ and standard deviation $\bar{\sigma}$
$[x_0, x_1, \ldots, x_L]$	L + 1bounds of the L intervals of a Interval Discrete Chance node of a BN
$f(x)$	Probability density function (pdf) of a random variable X
$\Phi(x)$	Cumulative distribution function (cdf) of a random variable X
σ_{min}	Standard deviation equals to half the interval in which a desired $\bar{\mu}$ falls
$q_i^{'}, q_i$	Probability of interval i of a non-periodic discrete node (not normalized and normalized respectively)
$p_i^{'}, p_i$	Probability of interval i of a periodic discrete node (not normalized and normalized, respectively)
Sub-section 5.1	
ACOPL3	Air changes per hour
AFlCNl, AFlCNop, AFlCNq, AFlSlb	Air flows in station's rooms named as CNl, CNop, CNq, Slb
AFlT31, AFlT32, AFlSF1, AFlSF2	Air flows in tunnels TF1 and TF2 and in station fan ducts SF1 and SF2
WiSMet	Outdoor wind speed
WiDMet	Outdoor wind direction
FreSF1, FreSF2, FreTF1, FreTF2	Frequencies that drive the two fans in the station and the two fans in the tunnels, respectively

http://dx.doi.org/10.5772/58470

DFreSF1, DFreSF2, DFreTF1, DFreTF2	Discretized frequencies that drive the two fans in the station and the two fans in the tunnels, respectively
PElSF1, PElSF2, PElTF1, PElTF2	Electric power absorbed by the two fans in the station and the two fans in the tunnels, respectively
TOuMet	Outdoor temperature
GaiTr1	Internal gain supplied by trains approaching on railway 1
NPeSta	Number of people in the station
TemPL3	Temperature in platform PL3
DTePL3	Variation of temperature in platform PL3
DTeOut	Variation of outdoor temperature
HFlOut	Heat flux coming from outdoor air
X_p01	Denotes one step ahead value for a variable X
Sub-section 5.2	
J	Cost function to be minimized by MPC
H	Prediction horizon
X_{Min}, X_{Max}	Minimum and maximum values allowed for a certain variable X
$\bar{X}$	Desired reference value for a certain variable X
$\tilde{X}$	Normalisation factor for a certain variable X
α	Weights of terms in cost function
X^+, X^-	Function that gets the absolute value of the positive and negative values of variableX, respectively

Acknowledgements

This work is part of the EU-funded research SEAM4US. Also, we are very grateful to our colleagues Engs. Roberta Ansuini and Sara Ruffini, who helped us develop the models.

Author details

Alessandro Carbonari*, Massimo Vaccarini and Alberto Giretti

*Address all correspondence to: alessandro.carbonari@univpm.it

Università Politecnica delle Marche, Department of Civil and Building Engineering and Architecture, Via Brecce Bianche, Ancona, Italy

References

[1] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, 2nd ed. Morgan Kaufmann Publishers, San Mateo, California; 1998.

[2] Levermore G.J. Building Energy Management Systems: Applications to Low-energy HVAC and Natural Ventilation Control, E & FN Spon, 2000.

[3] Di Hermann M., Hansemann T., Hobner C. Building Automation: Communication systems with EIB/KNX, LON and BACnet, Springer, NY; 2009.

[4] DALI Manual, available on-line at: http://www.dali-ag.org/c/manual_gb.pdf.

[5] Daum D., Morel. N. Assessing the total energy impact of manual and optimized blind control in combination with different lighting schedules in a building simulation environment. Journal of Building Performance Simulation, 3(1), p. 1-16, 2010.

[6] Oldewurtel F., Gyalistras D., Gwerder M., Jones C., Parisio A. Increasing energy efficiency in building climate control using weather forecasts and model predictive control. In: Clima-RHEVA World Congress Proceedings, Antalya, Turkey, 2010.

[7] Oldewurtel F., Parisio A., Jones C., Morari M., D. Gyalistras D. Energy Efficient Building Climate Control using Stochastic Model Predictive Control and Weather Predictions. In: Proceedings of the American Control Conference, Baltimore, MD, 2010.

[8] Mahdavi A., Orehounig K., Pröglhöf C.A simulation-supported control scheme for natural ventilation in buildings. In: Proc. of the 11th IBPSA Conference, Glasgow, Scotland; 2009.

[9] Henze G., Kalz D. Experimental analysis of model-based predictive optimal control foractive and passive thermal storage inventory, HVAC&R, 11(2); 2005; p. 189-213.

[10] Coffey B., Haghighat F. A software framework for model predictive control with GenOpt, Energy and Buildings. 42; 2010; p.1084-1092.

[11] Clarke J. A. et al. Control in Building Energy Management Systems: The Role of Simulation, In: Proccedings of the Seventh International IBPSA Conference, Rio de Janeiro, Brazil August 13-15; 2001.

[12] Maciejowsli J. M. Predictive Control with Constraints, Prentice Hall, England; 2002.

[13] Wetter, M. A Modelica-based model library for building energy and control systems. In: Proc. of the 11th IBPSA Conf., Glasgow, Scotland; 2009.

[14] Wetter M. Modelica Library for Building Energy and Control Systems: Buildings 1.0_build2; 2001. (accessed on: Dec 09, 2011), available on-line at: https://www.modelica.org/libraries/Buildings/releases/1.0_build2

[15] McGraw-Hill, editor. Concise Encyclopedia of Engineering. The McGraw-Hill Companies, Inc.; 2002.

[16] Qin S.J., Badgwell T.A. A survey of industrial model predictive control technology. Con. Eng. Practice 2003; 11(7) 733-764; Elsevier.

[17] Gyalistras D., Gwerder M., Oldewurtel F., Jones C.N., Morari M., Lehmann B., Wirth K., Stauch V. Analysis of Energy Savings Potentials for Integrated Room Automation, 2010. Clima-RHEVA World Congress, Antalya, Turkey, May 2010.

[18] Gyalistras D., Gwerder, M. Use of weather and occupancy forecasts for optimal building climate control (OptiControl): Two years progress report. Terrestrial Systems Ecology ETH Zurich, 2010. Switzerland and Building Technologies Division, Siemens Switzerland Ltd., Zug, Switzerland.

[19] Lehmann B., Wirth K., Carl S. Modeling of buildings and building systems. In: Proceedings of the 10th REHVA World Congress Clima 2010, 9-12 May 2010, Antalya, Turkey.

[20] Rodney A. Brooks The intelligent room project: The Second International Cognitive Technology Conference, pp. 271-279; 1997.

[21] Korb K.B., Nicholson A.E. Bayesian Artificial Intelligence. Chapman & Hall/CRC Edition; 2004.

[22] Spiegelhalter D.J., Dawid A.P., Lauritzen S.L., Cowell R.G. Bayesian analysis in expert systems. Statistical Science 1993; 8(3), 219–283.

[23] Heckerman D., Bayesian Networks for Knowledge Discovery (Chapter 11), in Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. Advances in Knowledge Discovery and Data Mining. MIT Press, Menlo Park/Cambridge/London; 1996.

[24] A.S.H.R.A.E. Guideline 14-2002. Measurement of Energy and Demand Savings. American Society of Heating, Ventilating and Air Conditioning Engineers, Atlanta, Georgia; 2002.

[25] Lazar M., Heemels W. P. M. H., Bemporad A., Weiland, S. Discrete-time non-smooth nonlinear MPC: Stability and robustness. In: Assessment and Future Directions of Nonlinear Model Predictive Control. Springer Berlin Heidelberg; 2007. p. 93-103.

[26] Aludaat, K. M., Alodat. M.T. A note on approximating the normal distribution function. Applied Mathematical Sciences. 2(9); 2008; p. 425-429.

[27] Mattsson S.E., Elmqvist. H. Modelica – An international effort to design the next generation modeling language. In: Boullart L., Loccufier M., Sven Erik Mattsson (editors). 7th IFAC Symposium on Computer Aided Control Systems Design, Gent, Belgium, April 1997.

[28] Wetter M. Modelica Library for Building Energy and Control Systems: Buildings 1.0_build2 (Dec 09, 2011), available at https://www.modelica.org/libraries/Buildings/releases/1.0_build2.

[29] Lawrence Hubert, Hans-Friedrich Köhn, Douglas Steinley. Cluster Analysis: A Toolbox for MATLAB. In: Roger E. Millsap, A. MaydeuOlivares. The SAGE Handbook of Quantitative Methods in Psychology. SAGE Publications Ltd; 2009. ISBN: 9781412930918.

[30] Ansuini R., Vaccarini M., Giretti A., Ruffini S. Models for the real-time control of subway stations. Proceedings. of: BS2013, Building Simulation for a Sustainable World, 13th International Conference of the International Building Performance Simulation Association, Chambery (F), August 2013.