Project acronym	B4B
Project full name	Brains for Building's Energy Systems
Grant No	MOOI32004
Project duration	May 1, 2021- December 31, 2025

Deliverable D1.05c

DBN-based FDD for Model Predictive Controls in energy-flexible buildings: what are the possibilities?

Lead beneficiary	TU Delft
Lead authors	Laure Itard
Publication date	31 December 2025
Deliverable Status	FINAL
File name	B4B WP1 D1.05c DBN-based FDD for suboptimal controls om energy-flexible buildings: what are the possibilities.docx

The project was carried out with a Top Sector Energy subsidy from the Ministry of Economic Affairs and Climate, carried out by the Netherlands Enterprise Agency. The specific subsidy for this project concerns MOOI subsidy round 2020

SUMMARY

Model Predictive Control (MPC) can be used to optimize building operation under dynamic behavior, and to account for uncertainties and constraints, for instance balancing comfort and energy efficiency, and incorporating external signals for grid interaction. MPC relies on the ability to predict the future (e.g. weather, occupancy patterns, prices, or even grid signals) in order to perform and coordinate timely actions necessary to ensure correct operation and energy-flexibility (like storing energy when it is cheap or the sun shines). Because of its high complexity and its dependence on correct predictions, MPC could be prone to many faults, resulting in suboptimal operation strategies, leading to higher energy use than intended, occupant discomfort and higher costs than expected. In this document we investigate the possibility to extend the existing 4S3F Bayesian Network-based Fault Detection and Diagnostic method to HVAC systems using MPC.

Even though diverse approaches have been used in MPC to cope with inherent uncertainties in the predictions, like stochastic and robust optimizations, reducing these uncertainties as much as possible remains essential for accurate predictions and optimal flexibility.

Following the 4S3F knowledge-based approach, possible faults were distinguished and aggregated to six main faults, belonging to each main part of the MPC controller: the disturbance model, the state function model, the optimization model, the reference state, the physical actuation in the HVAC system itself, and the sensors measuring the realized sate. For these faults, a total of seventeen symptoms were identified. Next to nine non-MPC-specific symptoms like broken sensors or actuators, eight MPC-specific ones were introduced. Five of them are symptoms that can be estimated during operation, by running the model co-currently, using recent historical data. The detection of some of these symptoms could also allow for automated regular recalibration of the models used in the MPC, or even to the automated choice of another optimization algorithm. For the other three symptoms off-line tests carried out on regular basis seem to be unavoidable.

Finally, the complete DBN for a MPC controller was built up and visualized. Thirteen symptoms were found to be each specific for one typical fault, while the remaining four symptoms could occur in relation to several faults.

Further research should demonstrate the validity of the approach in practical cases, including research on how many data from multiple time horizon periods must be stored to be able to conduct performance analysis while keeping EMS storage capacity acceptable. Demonstrating the approach on multi-objective MPC with multiple control variables is needed as well, while also considering different hierarchical configurations with multiple Model Predictive Controllers.

NOMENCALTURE

BES: Building Energy Simulation Model

CV: Control Variable

CVopt: Calculated optimal value of CV

DR: Demand Request

DSM: Demand Side Management

EMS: Energy Management System

FDD: Fault Detection and Diagnosis

GOF: Goodness-Of-Fit

HVAC: Heating, Ventilation and Air Conditioning

MAE: Mean Absolute Error

MPC: Model Predictive Control

NMBE: Mean Bias Error

PBIAS: Percent Bias

PID: Partial Integral Derivative Control

RMSE: Root Mean Square Error

SO: Stochastic Optimization

RO: Robust Optimization

4S3F: 4 Symptoms- 3 Faults Bayesian Network-based FDD

TABLE OF CONTENTS

1 INTRODUCTION

The Brains4Buildings (B4B) project aims for a part at developing fault detection and diagnostic methods (FDD) for Heating, Ventilation and Air Conditioning Systems (HVAC), as these methods can help reduce unnecessary energy consumption and discomfort in buildings by signalling timely broken components, biased sensors, wrong setpoints, poor tuning between components and systems and even poor assumptions relating to their operation.

In Work Package 1, the focus is on Diagnostic Bayesian Network methods, such as the 4S3F method, as they combine a knowledge-based and a data-driven approach in the most efficient way (see [1] for an extensive literature review). The knowledge-based approach ensures that knowledge embedded in HVAC system design (system architecture and control knowledge as well as in-depth component knowledge) is unlocked and directly used to construct the DBN ([2], [3], [4], [5]). For the fault diagnostic itself, the DBN uses as input millions of data points from the Energy Management System (EMS). This data can also be used in a data-driven approach to further enhance the performance of the DBN diagnostic by providing better insights in conditional probabilities between symptoms and faults, in occurrence of faults (prior probabilities), and in scalability of the DBN (can one DBN be reused for other similar HVAC systems in other buildings) ([6]). As reported in previous references and in [7], the approach was developed and tested in living labs.

At the same time, methods for energy flexibility have been researched in Work Package 2, in order to get more grip on the different ways to dynamically match the intermittent supply of renewable energy with building's energy demand. Different strategies can be applied like peak shaving by reducing and shifting demand, and by storage technologies at demand and supply sides. Dynamic energy pricing can be an incentive for shifting and storing strategies. Conventional control methods are typically designed for static operating conditions and fail to account for dynamic factors, while advanced control techniques, particularly Model Predictive Control (MPC), can be used to optimize building operation under dynamic behavior, and to account for uncertainties and constraints, for instance balancing comfort and energy efficiency, and incorporating external signals for grid interaction, like Demand Requests (DR). MPC relies on the ability to predict the future (e.g. weather, occupancy patterns, prices or grid signals) to perform and coordinate timely necessary actions (like storing energy when it is cheap or the sun shines).

The importance of energy flexibility, and therefore of MPC, is expected to increase rapidly the coming years. Because of its high complexity and its dependence on correct predictions, MPC could be prone to many faults, resulting in suboptimal operation strategies, leading to higher energy use than intended, occupant discomfort and higher costs than expected. In this document we investigate the possibility to extend the 4S3F-FDD method to HVAC systems using MPC.

In chapter 2, a short inventory of how MPC methods usually deal with prediction uncertainty is given, as too high uncertainties could lead to faulty (i.e. non-optimal) decisions and control strategies. In chapter 3 we propose a framework for the inclusion of MPC in Fault Diagnosis methods and in chapter 4 conclusions and recommendations are drawn.

2 DEALING WITH UNCERTAINTIES IN MODEL PREDICTIVE CONTROL

2.1 PID and Model Predictive Controls

PID (Proportional Integral Derivative) controllers are most generally used to control HVAC components (see Figure 1).They ensure that a desired state (Reference state, for instance a desired CO2-concentration level[1]) is achieved by controlling HVAC and/or building variables (e.g. fan speed to bring more or less outdoor air in a building) and comparing the realized state (the one measured by a CO2-sensor) to the desired one. A PID controller looks at the present time and considers the immediate past to decide on which action to undertake on the controlled variable (fan speed u in Figure 1). It reacts proportional to the magnitude of the difference between desired state and realized measured state and integral by considering the cumulative sum of past errors. The derivative part uses the rate of change of the error to limit overshoot. These controllers work smooth when changes in the desired state are not too sudden, when not to many state objectives must be realized concurrently, and when the variables controlling the states do not interact with each other.

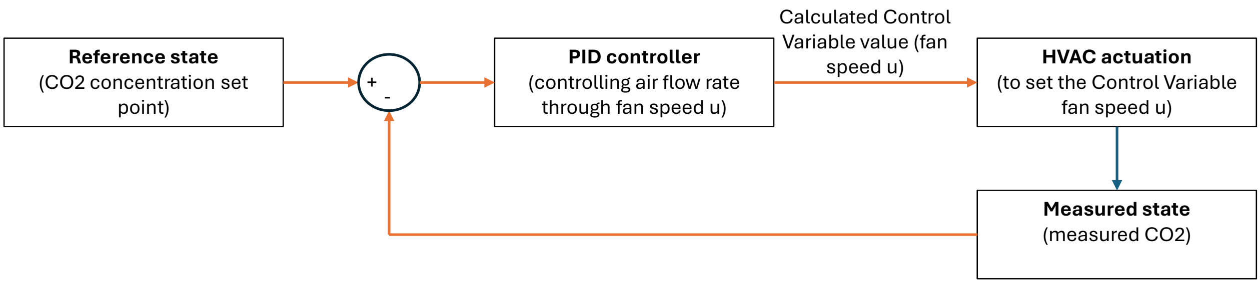

Figure 1: Schematic representation of the working a PID controller, applied to the control of CO2-concentration.

In systems where multiple state objectives must be achieved at the same time (e.g. low energy usage, low costs, low CO2-concentration, high comfort), or a variable is interacting on more states (e.g. fan speed in Figures 1 and 2 affects oppositely CO2-concentration and energy usage), or the state objectives can change suddenly depending on external variables (called Disturbances), like room occupancy, Model Predictive Control (MPC) is preferred, see Figure 2. For a basic understanding of MPC without mathematical formulations, see [8].

Figure 2 gives an example of MPC with only one state objective (CO2 concentration) and one control variable (fan speed u). MPC is about finding out which control steps can best be undertaken to ensure the reference state is smoothly achieved in future. It starts therefore from a model-based prediction of the future state objective, accounting for all important characteristics of the system, including current measured state and disturbances (occupancy in Figure 2, but could also be weather or prices). This model is then fed to an optimization algorithm searching for the best value of the control variables (fan speed u in Figure 2) to get the measured state as close as possible to the reference state. This is done for all time steps covering a chosen time horizon. Generally, a receding horizon is used, meaning that action on the control variable is taken only for next first time step (t+1). At next time step, optimization is carried out again over the new time horizon. This way MPC can adapt dynamically the actions taken and respond to dynamically changing circumstances.

[1] In this example, values of the desired CO2-concentrations would have been chosen to ensure both a healthy environment (no too high concentration) and low energy use (not too much ventilation, therefore not too low concentration). In the example we consider however one unique objective value (e.g. 850 ppm) instead of a range of values (e.g. 400-1200 PPM).

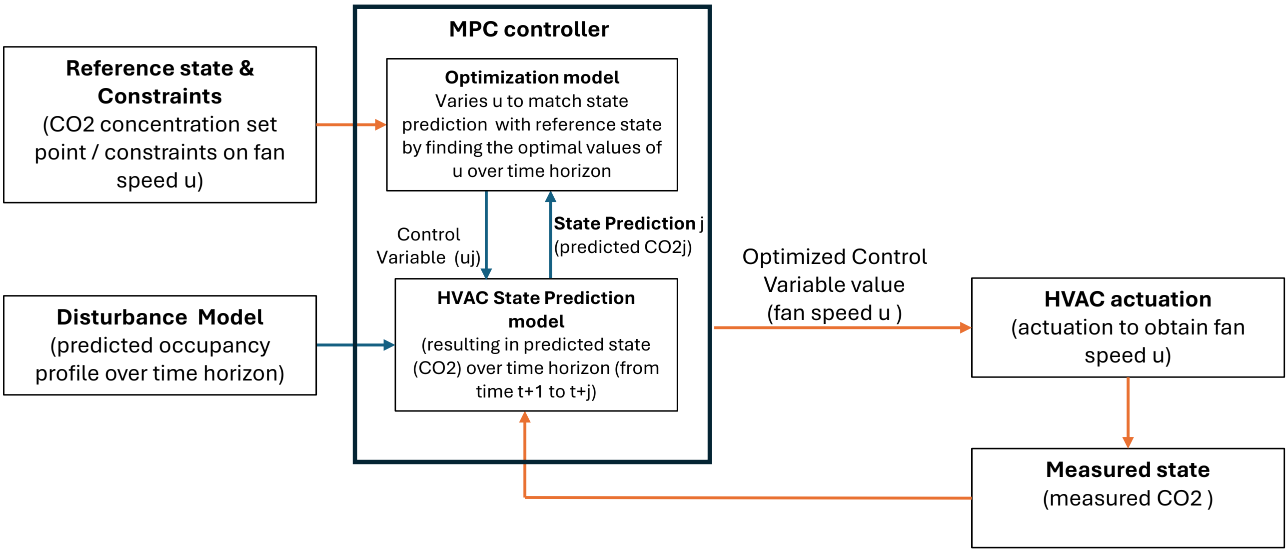

Figure 2: Schematic representation of the working a MPC controller, applied on the control of CO2-concentration.

In the case of multiple objectives, the objectives can be conflicting as trade-offs take place, and multiple solutions are present, depending on the weight given to the different objectives (e.g. Pareto optimization). For example one can want to minimizing the energy consumption while putting bounds on the costs and the indoor temperature (comfort). Very often there are also constraints on the control variables (e.g. a maximum speed).

2.2 Uncertainty Management in MPC

As extensively explained in [9], integrating MPC with Demand Side Management (DSM) for energy flexibility raises issues related to uncertainty management, as it is based on predictions of weather, prices, energy demand and supply. By nature, predictions are uncertain, and 'erroneous' predictions may lead to suboptimal control strategies. To ensure correct performance, uncertainties need to be properly taken into consideration.

To account for parameter prediction uncertainties in MPC, diverse approaches can be used such as (see [9] to [16]):

- Worst-case approach
- Stochastic optimization (SO)
- Robust optimization (RO)

Worst case approach

The worst-case approach is very robust, as the most adverse value of the predicted parameters is taken as a basis to optimize the objective function. For instance, in winter, if the outdoor temperature prediction is 7°C ± 2°C, the algorithm will use 5°C to find best heating strategies. It is a very robust approach, as in any case the needed heating power will be available, but also leads to suboptimal decisions, as most of time the actual temperature will be higher than the lower bound, and other strategies would have been more efficient (for instance heat storage in a buffer tank could have been applied).

Stochastic optimization and Robust Optimization approaches have been developed to mitigate these too conservative decisions.

Stochastic optimization

In stochastic optimization, it is typically assumed that the probability distribution of the uncertainties is known (e.g. gaussian), and the goal is to minimize the expected value of an objective function with respect to these uncertainties. While this SO approach can lead to less conservative decisions than the worst-case approach, analytical solutions exist only for limited classes of distributions and problems. Moreover, the true probability distribution of uncertainties is often unknown or difficult to estimate accurately. To overcome this limitation, scenario-based approximations of the stochastic formulation are often employed, where the uncertainty distribution is approximated by a set of uncertainty samples.

Robust optimization

Robust optimization represents an alternative and complementary approach. In the RO setting, uncertain parameters are assumed to lie within a prescribed uncertainty set, which captures all possible realizations. The objective function and constraints are then optimized and enforced against the worst-case realization within this set. Unlike the stochastic approach, robust optimization does not require explicit probabilistic information about the uncertainties – only their range. RO formulations are generally computationally efficient, and their computational cost does not depend on the number of uncertainty samples. However, a key challenge in RO lies in the construction of the uncertainty set. Data-driven RO frameworks have emerged as a promising direction. In data-driven RO, the uncertainty sets are constructed directly from empirical data rather than predefined assumptions. Within B4B, an algorithm was proposed by Li ([10], [12], [14], [15]) to generate data-driven uncertainty sets that are both compact and adaptive to irregular data distributions. Representing uncertainty sets as unions of multiple basic subsets has been shown to improve the performances of data-driven RO. Therefore, Distributionally Robust Optimization (DRO) was proposed to mitigate the conservatism of conventional RO formulation while avoiding high computational burden.

Necessity of Fault Diagnostic

While, as described above, different MPC formulations have been developed to cope with inherent uncertainties in parameter prediction, the quality of the results remains dependent on the quality of parameter prediction. For instance one can imagine that, when using solar radiation predictions from the nearest meteorologic station, the predictions can be systematically biased if (part of) the building is shadowed by trees or another building; Or if the nearest station is too far away, the local weather may just be different than at the station, or there can be a time delay. If there are no measurements close to the building, it will be difficult to generate uncertainty sets to account for this. On the same way, energy demand side prediction may be wrong if the opening times of the building change, holidays are not accounted for of if maintenance is carried out. It is known that after maintenance activities, set points may have been changed accidently, or to solve a local problem without looking at the repercussions on other parts of the system.

In the next chapter we propose an extension of the 4S3F FDD method to cope with fault diagnostic issues relating specifically to the use of MPC.

3 FDD FRAMEWORK FOR MPC-BASED DEMAND SIDE MANAGEMENT

3.1 Introduction

After a brief explanation of Fault Detection and Diagnosis using Diagnostic Bayesian Networks (4S3F-FDD) in section 3.1, an analysis of its application to MPC-based demand side management is conducted in sections 3.2 to 3.4. In section 3.2 the analysis of possible faults in the MPC is conducted, while in section 3.3, symptoms related to these faults are inventoried. In section 3.4, the complete DBN is constructed and briefly analysed.

3.2 3.1 Fault and symptom categories in 4S3F-FDD

In 4S3F-FDD 3 types of faults are distinguished: component faults (like a broken fan or biased sensor), control faults (like a wrong set point temperature) and model faults (like assuming no heat losses in ducts), see Figure 3.

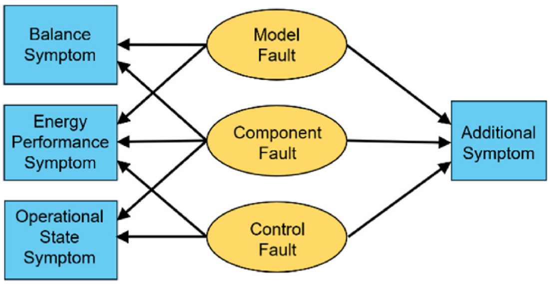

Figure 3: General setup of Bayesian network-based Fault Detection and Diagnosis (from [2])

To automatically identify these faults, it is necessary to (automatically) detect symptoms relating to the fault. Symptom detection depends on sensors: for instance, a CO_2-meter indicating a too high value may be a symptom of a broken fan, a wrong setpoint, or a bias in the sensor itself. Humans can also be used as sensors: complaints about the indoor climate (like too warm, too cold, draught) are symptomatic of certain faults in the HVAC system (category 'Additional symptoms' in Figure 3). In a DBN, symptoms are related to faults, indicating a known relationship between both. To identify the presence of a symptom, postprocessing of the sensor data is usually needed, for instance the comparison of a CO_2-meter reading with maximum value allowed or the comparison of control setpoint values with prescribed values (category 'Operational State Symptom' in Figure 3). The use of a model to estimate energy performance over a component or a system, based on temperature and volume flow rate readings for instance, will help to compare the actual performance with the expected one and is also an example of preprocessing for symptoms belonging to the categories 'Balance' or 'Energy Performance'.

In a DBN structure, symptoms are referred to as parent nodes, while faults are children, and their relationships are given in a graph, together with prior probabilities and conditional probabilities. For methods to estimate prior and condition probabilities, see [6] and [17].

3.3 3.2 Possible Faults in MPC

The performances of Model Predictive Control depend on the absence or presence of faults in the different parts of the MPC and HVAC system (see also Figure 4). One could state that a main problem in FDD for MPC-controllers is to make sure that the estimated optimized control variable value is really optimal, which is very difficult to trace. That is why it is useful to bring further differentiation of faults in the different parts of the controller.

Faults can be:

a) Incorrect working of the control variable (e.g. fan speed regulation) in the HVAC system

b) Incorrect working of the sensors measuring the state variable
c) Incorrect reference state and constraints
d) Poor quality of the disturbance prediction model and the data used in this model (e.g. the predicted occupancy)
e) Poor quality of the state prediction model (e.g. the model predicting CO2)
f) Poor quality of the optimization model

Note that options a, b and c are not specific to MPC and are valid as well in FDD for PID controllers. By 'Poor quality' we mean that the lack of quality is such that it compromises the accuracy of the calculated value of the optimized control variable and can therefore be considered as a fault.

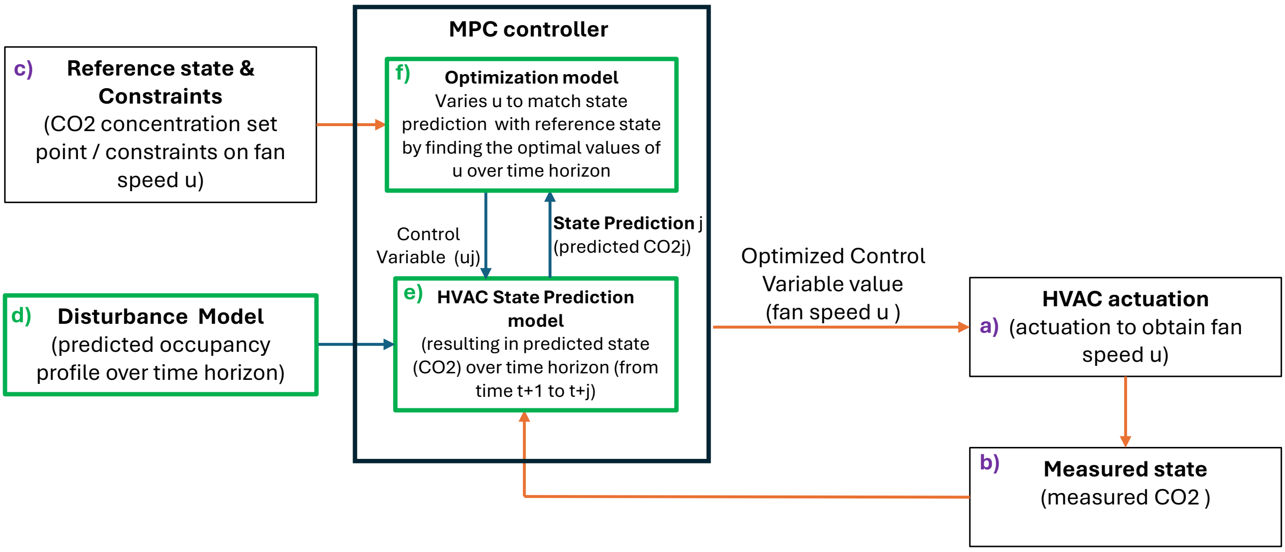

Figure 4: Possible faults in MPC (Green: Model faults; Purple: Control and component faults also arising in PID controllers)

a) Incorrect working of control variable (Component of Control fault)

The control variable (fan speed in Figure 4) may not respond to signals as expected, due to hardware and communication faults: the actuator can be stuck, the communication protocol can be down, the component itself can be broken (e.g. the motor). In this document we aggregate all this possible faults to one: Incorrect working of the control variable.

b) Incorrect working of sensors for state measurement (Component fault)

Sensors can be broken, too inaccurate for the purpose, or biased. As the MPC uses the sensor readings to steer, this will unavoidably lead to problems. Here too we aggregate the possible faults to one.

c) Incorrect reference state and constraints (Control fault)

The reference state itself may be wrong, i.e. not aligned with design documents, or not realistic: e.g. a temperature set point that remains constant over day and night, while the specifications indicate that the night setpoint should be lower at night; or a CO2 set point lower than outdoor CO2-concentration. Constraints may also be wrong (e.g. a maximum speed higher than technically possible with the fan). Note that in both PID control and MPC the reference state is generally a bandwidth rather than a single value.

d) Poor quality of the Disturbance Prediction Model (Model fault)

The accuracy of the models used to predict the future values of some input parameters (disturbances) to the state prediction model impacts the results of the MPC. For instance, predicting the presence of people in a room two hours later may lead to the decision to start heating the room already. But if the occupancy finally appears to be zero, another strategy could have been better (e.g. storing heat in a buffer tank instead of starting room heating). **If the accuracy of the prediction model falls under an acceptable (inherent) uncertainty band, smart optimization algorithms like RO or SO algorithms described in Chapter 2 will make the best of it. But narrowing the uncertainty bandwidth as much as possible will lead to more secure outcomes.**

A prediction model can be a simple regression model, a multi-linear or a quadratic one, a knowledge-based model etc., and should be able to catch variations of the future disturbances up to the simulation horizon. If the model is not suitable, the fit will impact the fitness of the MPC-selected solution.

Note that in some cases, the disturbance prediction model is reduced to just passing over data estimated elsewhere, like weather predictions from meteorological institutes or energy price developments from grid operators. It may be that the predicted meteorological data are right at the station level, but not at the building level (if the building is too far away from the station) or at the zone level (even if the meteorological station is placed on the roof of the building, solar radiation predictions on the facades may be inaccurate). The model can then be seen as faulty.

In any case, the prediction models for future disturbances generally depend strongly on historical and current data. **So the quality of this data is essential too.** Even when a data-driven model predicting the outdoor temperature is correct, if the data comes from a biased temperature sensor, in the end, all predictions will be wrong, affecting therefore the MPC. Many other types of prediction models (knowledge based for instance) also use data as input, e.g. for calibration or further processing.

Fault d) is therefore an aggregation of faults in the input data to the disturbance prediction model and faults in the disturbance prediction model itself.

e) Poor quality of State Prediction Model (Model fault)

If the state prediction model does not catch all important parameters, the performances of the MPC and demand side management will be impacted. As an example, if the objective function is the total heating energy consumption, and the model does not account for hot tap water, ventilation losses or solar heat gains, it will not describe reality well, and this will impact the chosen action, and therefore also the solution for demand side management. In the same way, if the model is a static one, while the time stamp is such that dynamic effects are relevant, problems can be expected. It is essential that the model describes with enough accuracy the relationship between the control variable (u in Figure 4) and the State Function (the CO2 set point in Figure 4)). As the state prediction model needs to make rapid real time calculations, it is often based on simplifications/linearization of more complex models, using Machine Learning algorithms to derive simpler relationships. This modelling step is done on beforehand, but here too faults can arise.

f) Poor quality of Optimization Model (Model fault)

Different optimization algorithms have different performances depending on the characteristics of the state function in the domain considered and the type of constraints. For instance is the function linear or not, is it constrained or unconstrainted, is it multi or single objective optimization? Some algorithms are also more sensitive to local minima than others. Some can cope with uncertainties in input data (see chapter 2), others not.

3.4 Symptom detection per fault and corresponding DBNs

To automatically detect the faults described above, they must be coupled to observable (detectable) symptoms. The problem could arise that there are little specific symptoms making it impossible to discriminate between faults in the 3 different models (disturbance, state prediction and optimization models). Even in this case it would be useful to at least know that something in the PMC controller is wrong. The symptoms that can be expected for each fault are described below.

Symptom detection for Fault a) (Incorrect working of the control variable)

As explained in section 3.2, this is a group of multiple faults, that can take place either with PID control or with MPC. A symptom (S1) for this group of faults can be that the control variable (e.g. fan speed u in Figure 4) does not achieve the setpoint value passed on by the MPC controller CVopt (e.g. Optimized u-value). In HVAC systems where the control variable itself is measured, a direct symptom would be a deviation larger than ε_u between the measured value and the setpoint value during the considered timestamp (or over multiple time stamps). For instance for the case of Figure 4 a symptom would be detected when Equation (1) is valid:

$$|u_{measured} - u_{optimal}| > \varepsilon_u \qquad (1)$$

It could also be that the calculated optimum CVopt is physically unrealistic, and cannot be achieved (S2).This would also indicate that the constraints in the optimization model are wrong. Another symptom could be a less direct one like the measured state (CO2 concentration in Figure 4) being systematically differing from the Reference State (S3).

Many other symptoms could support evidence as well (see for instance [6] and [18]): In our example, complaints about air quality (damp air, draught), comfort (too cold or too warm) could be symptomatic, as well as CO2 concentration measured as room level, depending on the HVAC system. They are aggregated in Symptom (S4). Figure 6 represents the corresponding DBN.

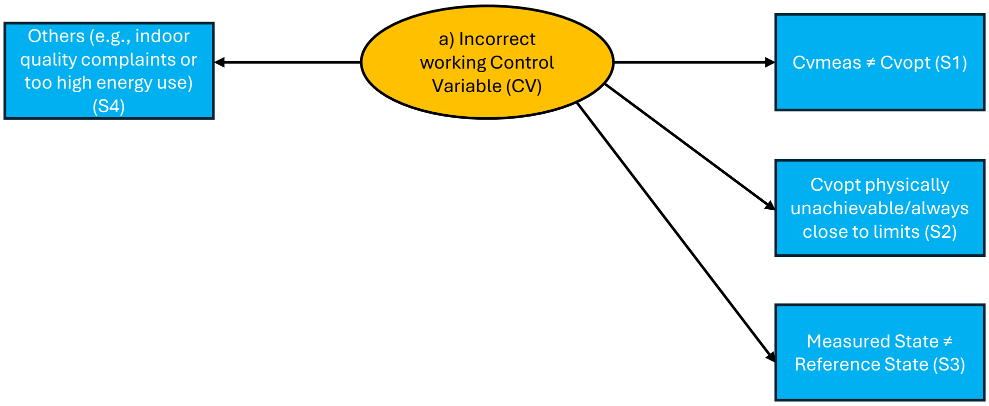

Figure 6: General DBN structure for Fault a) (Incorrect working Control Variable).

Symptom detection for Fault b) (Incorrect working of the sensor measuring the state variable)

Symptoms of a faulty state sensor can be a very constant value over time (no fluctuations at all (S5)), or an unrealistic value (S6) (e.g. a CO2 value above 5000 ppm, or below atmospheric concentration); In case of multiple similar sensors in similar rooms, deviations between them can also be used as symptoms for bias or poor calibration (S7). Here too, the symptom detection is identical with PID controllers or MPC. Many examples can be found in e.g. [3], [6] and [18]. Figure 7 pictures the corresponding DBN.

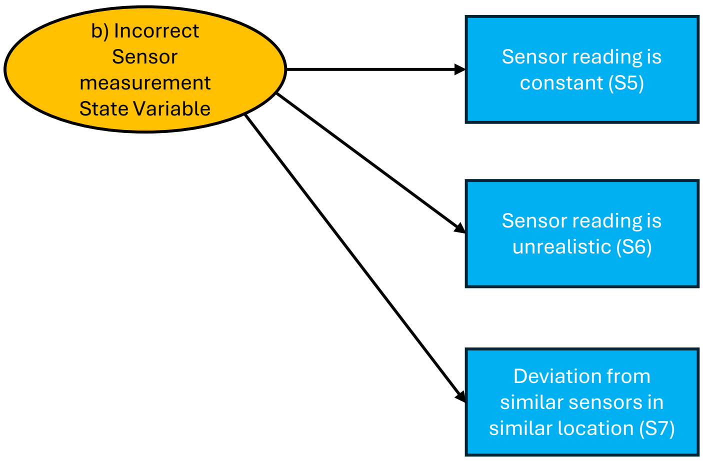

Figure 7: General DBN structure for Fault b) (Incorrect working of the sensors measuring the state variable).

Symptom detection for Fault c) (Incorrect reference state and constraints)

There are different ways to diagnose an incorrect reference state, see also Figure 8:

- Analysing deviations between the programmed reference state and the one from the technical documentation, common sense or literature (S8). This is similar to conventional PID controllers.
- Next to general symptoms as described in (S4), the analysis of the realized State function against historical data is recommended (S9). This can be done in parallel during operation. As an example, if the objective is to minimize energy costs, the energy import (from grid operator) can be plotted against energy tariffs over time. If the energy import is often high at high tariffs, this would be symptomatic of an error, either in reference state & constraints or in disturbance or optimization models. Another example, relating to Figure 4, would be the CO2 reference state being achieved when the building is unoccupied, and not being achieved during occupancy. Note that in some cases S4 and S9 may be identical, and should then be merged as one symptom.
- As MPC is also using constraints, CVopt not realistic or being very often close to the limits could be a symptom that either the constraints or the reference state are not realistic (S2).

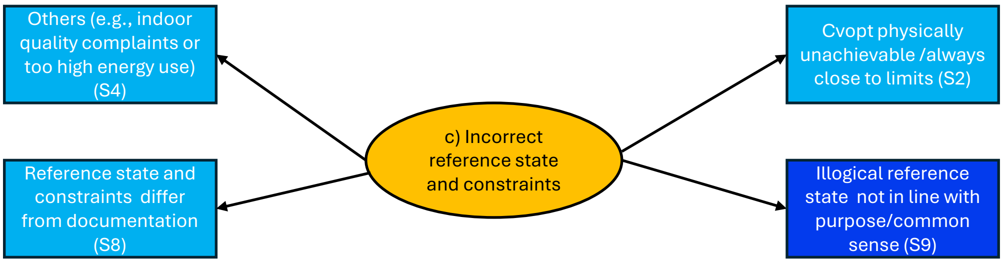

Figure 8: General DBN structure for Fault c) (Incorrect reference state and constraints). Dark blue: Symptom identification needs comparison with historical data.

Symptom detection for Fault d) (Poor quality of disturbance prediction model and data used in this model)

Indirect symptoms of this fault could be, similar to Fault a), that the measured state (CO2 concentration in Figure 4) differs systematically from the Reference State (S3). Also complaints about air quality or comfort could be symptomatic (S4), as well as illogical actual states (S9) and CVopt not being realistic or being often close to the limits (S2). However, these symptoms are quite general, and lack specificity to lead to Fault d).

It may be difficult to find specific direct symptoms for such fault, as the fault may have strong cascading effects all over the control chain: an inaccurate disturbance prediction model will most probably lead to inaccurate results of the state function model, and therefore to suboptimal decisions by the optimization model, which would lead in turn to 'wrong' setpoints of the control variable CV (optimal u in Figure 4). In some cases, this could lead to a measured state not achieving the Reference One, as noticed above.

Because of the lack of direct specific symptoms, specific additional tests to support symptom detection for both the disturbance prediction model and its input are investigated in the next paragraphs.

Disturbance Prediction model

Let's take the example of weather predictions: based on current and past weather the model predicts future weather over the chosen time horizon. By storing these predictions in the EMS and systematically comparing them later with the really achieved values at these time stamps, deviations between reality and prediction can be identified. Most probably, the same prediction accuracy is not needed throughout the time horizon, as in receding horizons the first values have more weight in decision-making (only the first step is implemented) than the values at the end of the horizon. Therefore, the further in the time horizon, the larger the acceptance for deviations could be. This is asking for an analysis per type of disturbance and per type of model: what time period is needed, and how large is the influence on the outcome of the state prediction model? What can be solved by SO and RO and what is out of range? Is there a bias in the disturbance prediction model (e.g. prediction is always higher or always lower)? If these problems are automatically detected during operation, automatic correction could be carried out by systematic recalibration of the model after a time period of a few time horizons - assuming that the model in use is suitable for the purpose. This can be done by continuous testing of performance parameters like RMSE, NMBE, MAE, PBIAS, GOF etc., which will also help to generate uncertainty samples for RO and SO optimization algorithms. Poor results of the disturbance prediction model on the performance parameters will be referred as Symptom (S10).

Data

As the source of data for predictions and for actual values is identical (for instance KNMI or a weather station on the roof; or in the case of occupancy, a predefined time table) there is a certain risk for self-referencing, and the approach described above cannot distinguish for the data not being the right ones. This could be the case when:

– Local sensors are biased or broken (S11). This may be detected by usual methods, as described under Fault b).
– Data are not fit-for-purpose ([19], [20]) e.g. weather data come from too far away (e.g. KNMI station not representing the local weather) or data is not up to date (e.g. timetable has not been updated timely). Solutions to identify this problem entail being aware of it (e.g. consult experts) or introduce hardware giving the possibility to calibrate the data (e.g. using an additional outdoor temperature sensor to compare with KNMI data, or adding occupancy sensors to compare with time tables). If this is not sufficient, it may be necessary to invest in more fit-for-purpose data collection. As the symptom 'Data not fit-for-purpose' (S12) cannot be identified during operation (except that S2 and S11 could contribute to indirect evidence), its **needs specific tests with specific additional sensors**, and is indicated in purple in Figure 9.

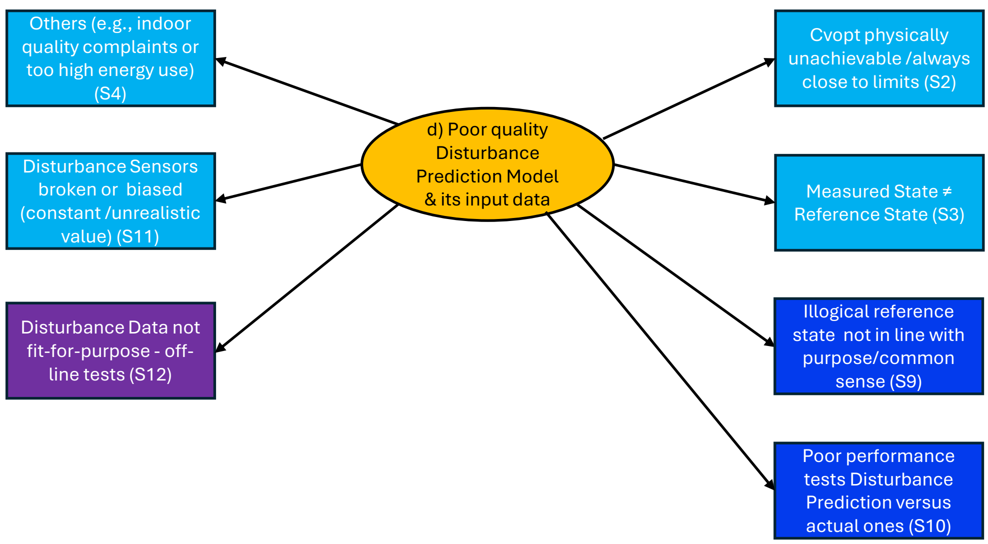

Figure 9: General DBN structure for Fault d) (Poor quality of disturbance prediction model & its input data). Dark blue: Symptom identification needs comparison with historical data; Purple: off-line tests needed.

Symptom detection for fault e) (Poor quality of the state prediction model)

A poor quality of the state prediction model will lead to a faulty or too inaccurate relationship between the control variable CV (u in Figure 4) and the state function (CO_2 concentration in Figure 4). Just like for Fault d), SO or RO algorithms can take care of inherent uncertainties and somehow compensate for a poor quality of the state prediction model. However, here too, increasing the accuracy will lead to more robust outcomes of the optimization model.

Symptoms as already described in Faults d) and a) are likely to appear, like not achieving the reference state (S3), CVopt not realistic or close to limits (S2), others (S4) and illogical actual states (S9).

Just as in the case of data faults in Fault d), it seems difficult to find specific symptoms that can be detected during operation. One approach could be **to run on regular basis and in parallel a detailed and validated HVAC/Building simulation model** (BES) and to compare its results with the results of the State Prediction Model for specific representative cases during the time period considered (at list a few time horizon periods). Poor agreement of the results (as measured through traditional performance indicators like RMSE) would be symptomatic (S13) for an incorrect state prediction.

Additionally, here too, **specific experiments (during system acceptance tests and/or regular maintenance moments) could be carried out** to regularly test and correct the state prediction model. This could include testing the difference between the measured reference state and the expected one for different settings of the control variable while the optimization model is switched off. A large difference between measured and calculated reference states would be symptomatic (S14) for a poor state prediction model. The model can then be improved by analyzing possibly missing parts or wrong equations, and carrying out related improvements, including calibration activities. Depending on the state parameters, this may need to be repeated at regular intervals to account for seasonal or week/weekends effects.

In both cases (parallel analysis and specific tests), calibration factors may play a role as easy-to-use correction factors between actual and calculated performances. Figure 10 summarizes the DBN structure for this fault.

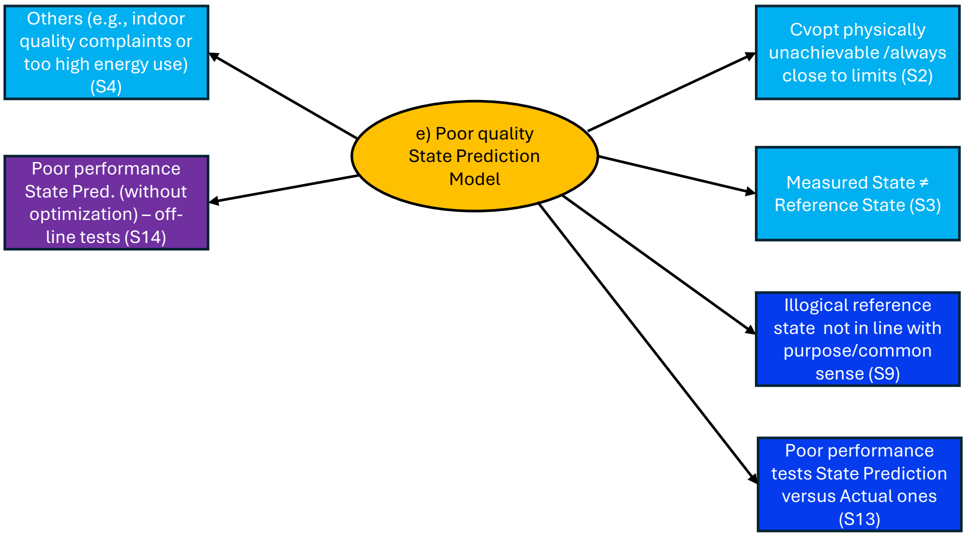

Figure 10: General DBN structure for Fault e) (Poor quality of state prediction model). Dark blue: Symptom identification needs comparison with historical data; Purple: off-line tests needed.

Symptom detection for fault f) (Poor quality of Optimization Model)

The chosen optimization algorithm must be suitable for the optimization problem considered. Choosing the right algorithm is work that is done before implementing the MPC, and also includes the settings of hyperparameters (if any), constraints and considerations about running time versus accuracy, as the optimization must be performed real-time. Correct choice of hyperparameters will help to deal with the presence of local minima's by inclusion of smart space investigation methods (e.g. grid or random search, Bayesian optimization). The question here is how to discover poor working of the model during operation: As the data fed to the model come from real world, unexpected changes and shifts could take place, additional unforeseen complexity could appear by which the optimization model would not be fit-for-purpose anymore.

Here too, symptoms like not achieving the reference state (S3), CVopt not realistic or close to limits (S2), others (S4) and illogical actual states (S9) could appear. It may also be that the running time increases strongly and gets close to the maximum one, and therefore convergence is not achieved in time (S15).

Additional tests could be considered to obtain more specific symptoms, by tunning hyperparameters on historical data (from the recent past) to test their validity and adapt them. This could be carried out in parallel mode (and therefore be automated) or in separate tests out of the system itself. Parallel mode is referred to as 'adaptive MPC' in section 7.4 of [21]. In terms of symptoms the results of such tests are that the hyperparameters are not right (S16). Finally re-analyzing off-line on regular basis the fitness of the optimization model with historical data would help to trace a lack of fitness of the model (S17). The corresponding DBN structure is given in Figure 11.

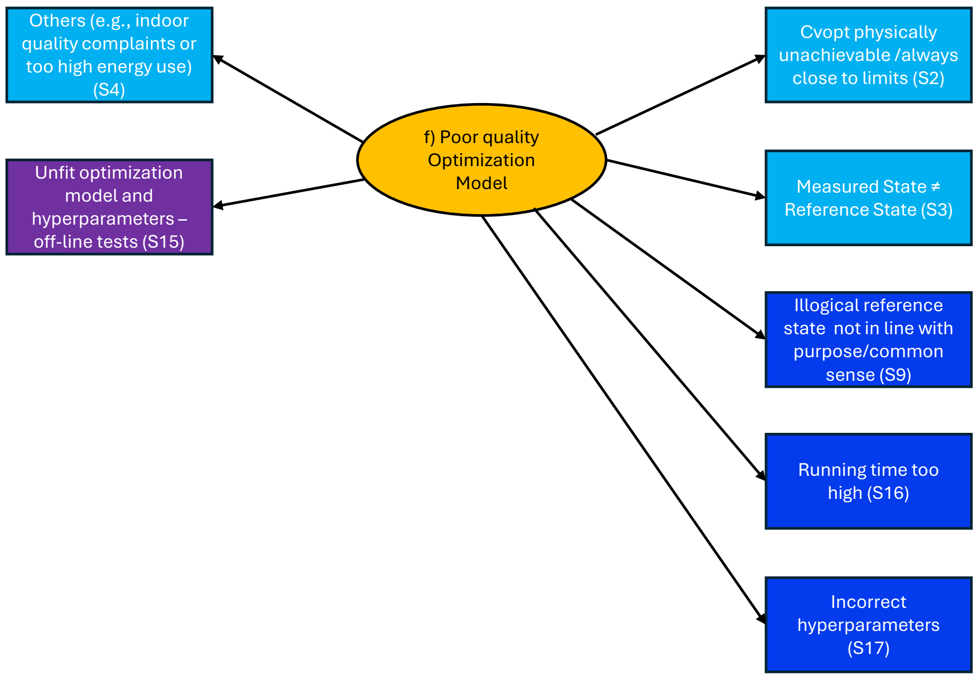

Figure 11: General DBN structure for Fault f) (Poor quality of optimization model). Dark blue: Symptom identification needs comparison with historical data; Purple: off-line tests needed.

3.5 Complete DBN structure

The complete DBN for the six faults identified around MPC, and the related 17 symptoms, is constructed from the separate DBNs and shown in Figure 12.

The left and right sides of the figure show the symptoms that are fault specific, and therefore relate to only one fault. If such a symptom is detected, it will lead to the identification of the specific fault with relatively high posterior probability. For instance Symptom S11 will lead to fault d) and indicate that the disturbance data or model are wrong. The same way S5 will lead with high certainty to fault b), and if S5, S6 and S7 appear together, there is little doubt that fault b) (Sensor measurement fault of the state variable) is present. All faults in this DBN have such fault-specific symptoms, which is very helpful for the diagnose.

The symptoms on the left part all relate to traditional symptoms in FDD for HVAC and can be detected by relatively straightforward comparisons between sensor readings and expected values and behaviour. Oppositely, most of the symptoms on the right part require more efforts to be detected: **either co-current calculations and performance analysis on near-historical data are needed, or even specific tests off-line are needed on regular basis, similar to acceptance tests or mandatory maintenance activities.**

The middle part of Figure 12 shows the symptoms relating to more faults. Symptoms S2, S3, S4 and S9 all four lead to faults d), e) and f), and therefore do not allow to discriminate between them. These are however useful symptoms as they show that the fault is occurring in the MPC controller or the disturbance model, rather than elsewhere (see Figure 4). In combination with the occurrence of other symptoms, more specific diagnostic is possible. Assuming similar prior fault probability for all faults and similar conditional probability between fault and symptom, one can see that, for instance if S1 is present together with S2, it is more probable that fault a) occurs than fault e). Oppositely, if S2 is present together with S14, fault e) is more likely to appear. Also interesting to note is that the off-line tests may be redundant with co-current performance analysis: each of the faults d), e) and f) is related to both types of symptoms. Observing both gives stronger evidence, but it may be that in practice co-current performance analysis would be sufficient to lead to a diagnose.

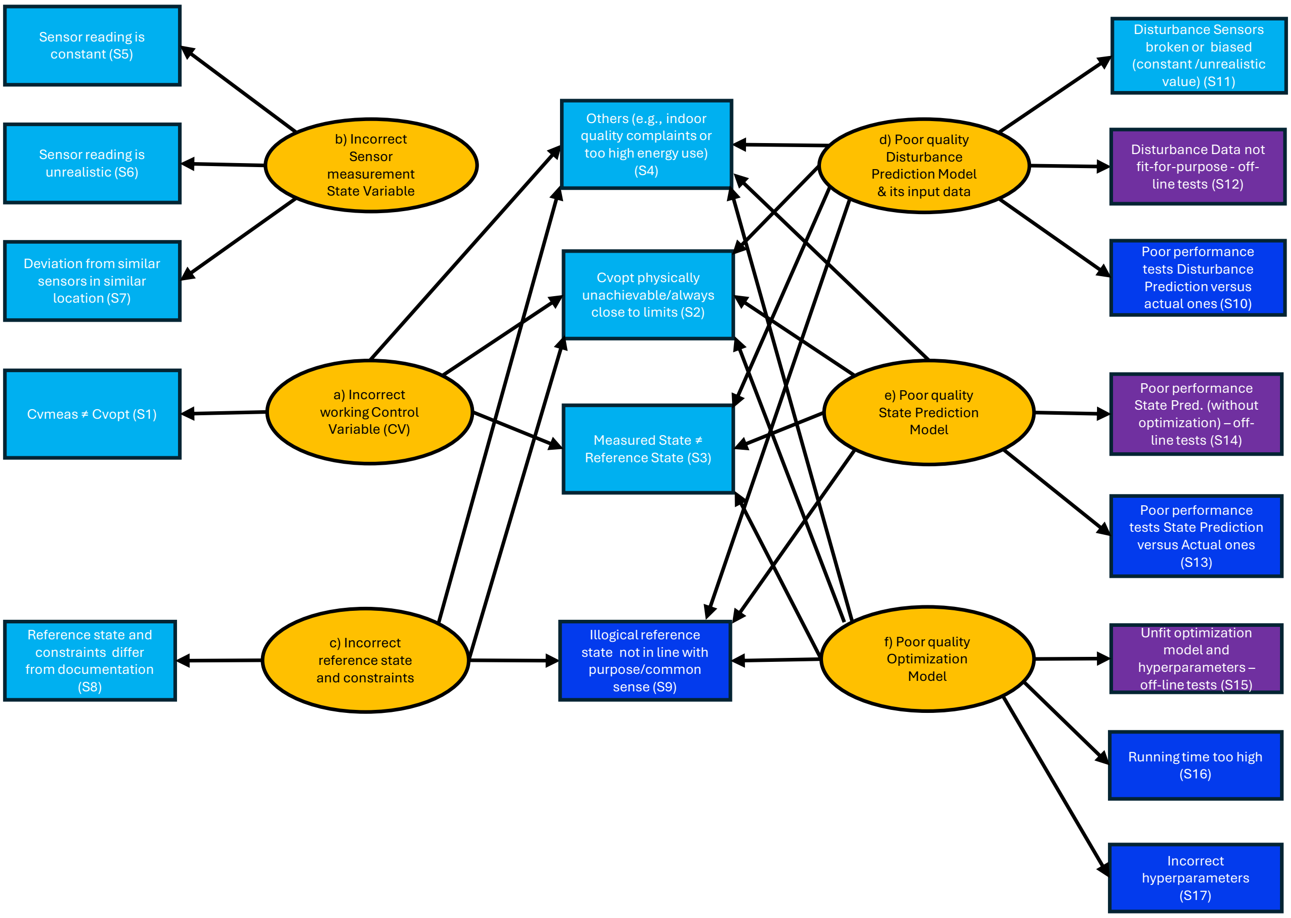

Figure 12: Complete DBN structure for FDD in MPC. Yellow: fault nodes; Light blue: non MPC-specific symptoms; Dark blue: Symptom for which identification needs comparison with historical data; Purple: Symptoms for which identification needs off-line tests.

4 CONCLUSIONS, LIMITATIONS AND RECOMMENDATIONS

4.1 Conclusions

In this research, we have shown how to extend DBN-based fault diagnostic (4S3F) to Model Predictive Controllers. Based on an analysis of the diverse components of MPC, 6 possible faults and 17 symptoms were identified and related to each other in order to construct a DBN. Next to 9 non-MPC-specific symptoms, 8 MPC-specific ones were introduced. Five of them are symptoms that can be estimated during operation, but the calculation needs to be made in parallel, using recent historical data. The detection of some of these symptoms could also allow for automated regular recalibration of the models used in the MPC, or even for the automated choice of another optimisation algorithm. For the other three symptoms, off-line tests on a regular basis seem to be unavoidable, although they may prove in later research not to be strictly necessary for an accurate diagnosis.

4.2 Limitations and recommendations

Four main limitations of this research have been identified:

- Although the research gives a first idea of how to set up DBN-based FDD for MPC, the approach has been purely theoretical, and the diagnostic framework was not tested. Further research should demonstrate the validity of the approach in practical cases.
- In this research we did not address the computational burden and the needed storage capacity to conduct on regular basis co-current tests around performance analysis of the diverse models used in the MPC. Further research is needed on how much data from multiple time horizon periods must be saved and stored to be able to conduct performance analysis without putting too much stress on EMS storage capacity.
- In the given examples we focussed on single objective, while most MPCs deal with multiple objectives and multiple control variables, therefore with multiple subsystems. Multiple objectives can be brought back to a cost function using weights for the different objectives, but demonstrating this in practice would be needed.
- Finally, we considered an MPC controlling one system, while in complex systems, multiple MPCs will be controlling multiple systems in different hierarchical configurations, see for instance Figure 13 from [21], in which a complete analysis of PMC configurations and methods for buildings and HVAC is given. The DBN approach for these configurations will need substantial refinements because of the multiple interactions between systems and controllers.

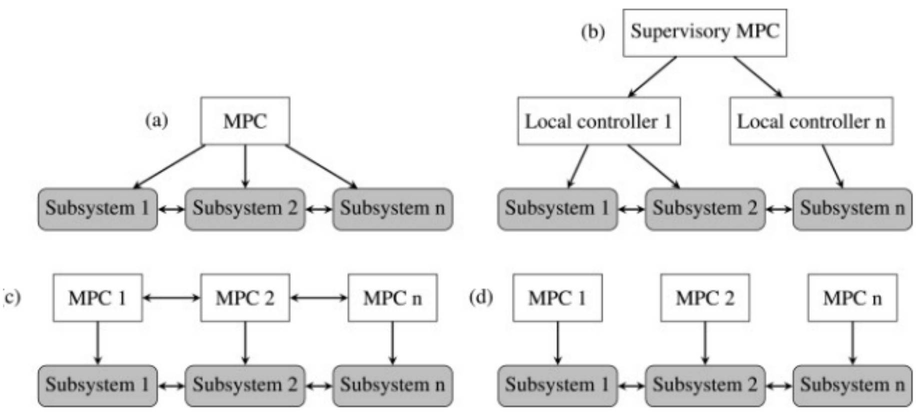

Figure 13: Diverse configurations of multiple MPC-controls: centralized (a); hierarchical (b); distributed (c); ; decentralized (d). Figure taken from [21].

REFERENCES

1) C. Lu, Z. Wang, M. Mosteiro-Romero, L. Itard, Diagnostic Bayesian network in building energy systems: current insights, practical challenges, and future trends, *Energy & Buildings 341 (2025) 115845*, https://doi.org/10.1016/j.enbuild.2025.115845 [B4B Publication]
2) Taal A.C., Itard L., 2020, P&ID-based automated fault identification for energy performance diagnosis in HVAC systems: 4S3F method, development of DBN models and application to an ATES system; *Energy & Buildings 224 (2020) 110289*
3) Taal A.C., Itard L., 2020, Fault detection and diagnosis for indoor air quality in DCV systems: Application of 4S3F method and effects of DBN probabilities; *Building and Environment 174 (2020) 106632*
4) Taal A.C., Itard L., 2020, P&ID-based symptom detection for automated energy performance diagnosis in HVAC systems, *Automation in Construction, Volume 119, 2020, 103344, ISSN 0926-5805,*
5) *https://doi.org/10.1016/j.autcon.2020.103344*
6) Lars van Koetsveld van Ankeren, Chujie Lu, Laure Itard, Implementing diagnostic Bayesian networks for heat recovery ventilation in real-world scenarios: A Dutch case study, *Journal of Building Engineering, Volume 111, 2025*, https://doi.org/10.1016/j.jobe.2025.113527 [B4B Publication]
7) Ziao Wang, Chujie Lu, Arjen Meijer, Shalika Walker, Laure Itard, Fault detection and diagnosis for heat recovery ventilation using 4S3F method: Impact of diverse sensor configurations, *Energy and Buildings, Volume 354, 2026*, https://doi.org/10.1016/j.enbuild.2025.116913 [B4B Deliverable D1.04c]
8) S. Gopalan, A. Rijs, S. Chitkara, A. Thamban, R. Kramer, Fault prioritization for air handling units using fault modelling and actual fault occurrence data, *Energy and Buildings 319 (2024) 114476*. https://doi.org/10.1016/j.enbuild.2024.114476 [B4B Publication]
9) Melda Ulusoy, MathWorks: *Understanding Model Predictive Control.* Part 1: Why use Model Predictive Control; Part 2: What is MPC?; Part 3: MPC Design Parameters; Part 4: Gain-Scheduled and Nonlinear Model Predictive Control
10) Yun Li, Data-Driven and Robust Predictive Control and Optimization, with Applications to Building Energy Management, *Thesis, May 2026, TUDelft*. ISBN 978-94-6518-276-6 http://repository.tudelft.nl/ [B4B Publication]
11) Li, Y., Yorke-Smith, N., & Keviczky, T. (2026). On data-driven robust optimization with multiple uncertainty subsets: Unified uncertainty set representation and mitigating conservatism, *Journal of Process Control, 158, 103611* [B4B Publication]
12) Tang, W., Li, Y., Walker, S., & Keviczky, T. (2025). Model Predictive Control for Unlocking Energy Flexibility of Heat Pump and Thermal Energy Storage Systems: Experimental Results. *Submitted to journal*, under review [B4B Publication]
13) Li, Y., Yorke-Smith, N., & Keviczky, T. (2024). Machine learning enabled uncertainty set for data-driven robust optimization. *Journal of Process Control, 144, 103339* [B4B Publication]
14) Tang, W., Li, Y., Walker, S., & Keviczky, T. (2024). Model Predictive Control Design for Unlocking the Energy Flexibility of Heat Pump and Thermal Energy Storage Systems, *2024 IEEE Conference on Control Technology and Applications (CCTA) (pp. 433-439), IEEE* [B4B Publication]
15) Li, Y., Yorke-Smith, N., & Keviczky, T. (2024). Robust Optimal Control With Binary Adjustable Uncertainties. *2024 European Control Conference (ECC) (pp. 3721-3727), IEEE* [B4B Publication]
16) Li, Y., Yorke-Smith, N., & Keviczky, T. (2023). Robust optimal control with inexact state measurements and adjustable uncertainty sets. *IFAC-PapersOnLine, 56(2), 562-569* [B4B Publication.
17) Li, Y., Yorke-Smith, N., & Keviczky, T. (2023). Unlocking energy flexibility from thermal inertia of buildings: A robust optimization approach. *2023 62nd IEEE Conference on Decision and Control (CDC) (pp. 2555-2562), IEEE* [B4B Publication]
18) Taal, A. Open library of standardized DBN models for the most common components & sensors, models and controls in HVAC installations, *B4B Deliverable D1.03a, April 2025* [B4B Publication]
19) Mosteiro-Romero, M., & Itard, L. (2025). Human-informed Building Automation: Enhanced Whole-Building System FDD. *REHVA European HVAC Journal* , *62*(6*), 41-45*. https://www.rehva.eu/rehva-journal/chapter/human-informed-building-automation-enhanced-whole-building-system-fdd [B4B Publication].
20) Xu L. Design optimization of seasonal thermal energy storage integrated district heating and cooling system: a modelling and simulation approach, Thesis TU/e, September 2020

21) Gaetani I, Mahdavi A., Berger C., Hoes P.J., Hensen J, Fit-for-Purpose Occupant Modeling: Choosing the Right Approach, in *Occupant-Centric Simulation-Aided Building Design: Theory, Application, and Case Studies, Routledge Book, 2023,* Fit-for-Purpose Occupant Modeling | 7 | Choosing the Right Approach |

22) Drgoňa J., Arroyo J., Cupeiro Figueroa I., Blum D., Arendt K., Kim D., Perarnau Ollé E., Oravec J., Wetter M., Vrabie D.L., Helsen L., All you need to know about model predictive control for buildings, *Annual Reviews in Control, Volume 50, 2020, 190-232, https://doi.org/10.1016/j.arcontrol.2020.09.001 .*

23) (https://www.sciencedirect.com/science/article/pii/S1367578820300584)