AN ALGORITHM FOR BEHAVIOUR PREDICTION OF COMPLEX TECHNICAL SYSTEMS

Osman, Krešimir (1); Štorga, Mario (2); Marjanović, Dorian (2)
1: Termo Servis Ltd., Zagreb, Croatia; 2: University of Zagreb, Croatia

Abstract

The objective of the research presented in this paper is to develop an algorithm for predicting behaviour of complex technical systems in an uncertain working environment. System's dynamic behaviour modelling and simulation should help to develop new and improve existing architectures of complex technical systems by mapping in both directions, from the structural to the behavioural domain and vice versa. The algorithm for predicting behaviour during system's architecture development consists of several operations shortly described in this paper. The proposed algorithm was verified on an example of a complex technical system – an air-handling unit.

Keywords: Algorithm, Behaviour, Prediction, Technical system, Uncertainty

Contact:
Dr. Krešimir Osman
Termo Servis Ltd., Zagreb
Technical Department
Croatia
kosman@fsb.hr

1 INTRODUCTION

Based on its fundamental principle (i.e. improving the quality of a product by minimizing the effects of variation without eliminating the causes), robust design has become a powerful tool to assist designers in making reliable decisions under uncertainty (Du, Chen, 2001). The aim of a robust design is not to try and eliminate the variations in product behaviour but rather to make the product insensitive to them. In developing efficient methods for environment uncertainty analysis, many researchers have been focused on design problems with a single discipline or an all-in-one integrated analysis (Du, Chen, 2000).

This paper presents an algorithm comprising several operations (methods) associated with the modelling of structural and behavioural domain of the technical system. The algorithm should help develop new and improve existing complex technical system architectures by bilateral mapping, from the structural to the behavioural domain and vice versa. In addition to using existing modelling methods, the algorithm allows predicting behaviour in an operating environment, also including uncertain working conditions. Based on information obtained by valuing the behaviour of complex technical system architectures, rules of behaviour of such technical systems are created and recorded. The presented algorithm is of an iterating nature, which means that several steps need to be performed through certain operations by using loopbacks in the structural and behavioural domains of the system to obtain the intended complex technical system architecture.The algorithm should help designers resolve issues regarding the development of complex technical system architectures more quickly and efficiently and designed to be used in initial stages of product development.

The next section briefly describes the state of the art and addresses the previous research in connection with the field of domain mapping. The research presented in this paper is an extension of such research. Section 3 motivated us to continue this research, while Section 4 provides a brief description of the proposed algorithm including all operations, explained on a step-by-step basis. Section 5 offers a verification of the approach on an example of an air-handling unit. A discussion of the results obtained on such example is provided in Section 6. Section 7, which closes this paper, offers a conclusion of the presented research and possible directions for future research.

2 STATE OF THE ART

Matrix methods are often used in the modelling of a system's structure. The most commonly used matrix approach in the literature is *DSM* (*Design (Dependency) Structure Matrix*), defined by *Steward* in his publication (Steward, 1981). DSM may also be used to decompose a system and analyse it qualitatively (Pimmler, Eppinger, 1994).On the other hand, *Hybrid Dynamic Systems* (Buss et al., 2002) are often used in system dynamics modelling. The dynamics of systems may be described as aligning and combining continuous and discrete domains. This enables modelling a full system range. However, both of these approaches are limited in their applications. A structural model cannot describe system behaviour while a hybrid dynamic system model cannot present interactions within the system and is not applicable to large and complex systems (Diepold et al., 2010). *Diepold et al.* proposed a new approach – *Multi-dynamic Mapping* (Diepold et al., 2010), which combines structural models and system behaviour model and presents them in a single system model. This approach also allows for both complexity types to be considered separately through different models. This approach was later extended to its environment referred to as *quad-I/HS* (*Interactive Interaction Importance Identification/Hybrid State*), (Diepold, Winkler, Lohmann, 2010).The approach uses feedbacks to the structural model to identify the acceptable subsystems for further analysis. In other words, these feedbacks adjust the system structure.

3 MOTIVATION

The deficiencies identified in the presented *quad-I/HS* approach motivated the author to propose an algorithm for developing complex technical system architectures. The central feature of the proposed algorithm is that offers the possibility of creating a mathematical model of each subsystem's behaviour (in a non-stationary condition), that is, presenting the change in the observed variable over time. Its other characteristics are as follows:

- the possibility of including a mathematical model of natural uncertainty (Kota, Chakabarti, 2009)in the mathematical model of observed technical system's behaviour,

- presenting the technical system behavioural model in a nonlinear form (Vukić, Kuljača, 2005), with the possibility of translating it into a linear form, except where the observed system is highly nonlinear and should be observed as such,
- the opportunity to improving technical system behaviour directly through the behavioural domain, by changing the parameters in the system behaviour mathematical model and by adjusting the MPC method parameters (Camacho, Bordons, 2004),
- the possibility of developing improved technical system architectures based on a similar behaviour model by using the MPC method, where behaviour is simulated according to a known variable change trajectory,
- the possibility of testing the stability of each process and subsystem within the tested technical system,
- recording the behaviour of a mechanical system or its variants by creating fuzzy rules,
- the algorithm is applicable to complex technical systems.

Some of the deficiencies in the *quad-I/HS* approach described were attempted to be rectified in the *Osman et al.* paper (2011) presenting the beginning of research relating to the Behaviour Prediction Framework in developing complex technical system architectures. These deficiencies in quad-I/HS approach were: impossibility to present mathematical model of the system and their subsystems, system representation in linear form and impossibility to make system improvement in the behavioural model. The research continued in the *Osman et al.* paper (2014) where the approach was extended and applied to a group of so-called nonlinear systems (the verification was carried out on an example of a load lifting device).

4   DESCRIPTION OF THE ALGORITHM

The complex technical system architecture development algorithm is presented in the form of a flow chart in Figure 1. The flowchart presents the course of operations and their execution in the algorithm through the structural and behavioural domains of the observed technical system. Provided below is a presentation of the algorithm, including a brief description of each operation. As shown in Figure 1, the algorithm is of an iterating nature. This means that sometimes several steps need to be performed through individualtransactions by using loopbacks in the structural and behavioural domains of the system. This will provide the desired complex technical system architecture, which could, subject to defined operating conditions, also operate under environment uncertain conditions. It consists of several operations briefly described in the following sentences:

1. The algorithm begins with an operation in the structural domain describing the initial system architecture. It is defined on the basis of the customer needs, the engineering requirements derived from them and the catalogue data of similar products. System components are defined on the basis of the created function. On the other hand, features are designed on the basis of product definition stage mention before.
2. The modularization of system architecture operation is used for description of initial system architecture (identification of possible modules). For this purpose component-based DSM (Design Structure Matrix) and a clustering process, (Pimmler, Eppinger, 1994) is used.
3. Such identified possible subsystems (modules) with defined initial assumptions (based on engineering requirements and designer experience) are used for creation of the mathematical models of system behaviour (for both behavioural domains - continuous and discrete).Also, these models are simplified due to simplifications on the basis of assumptions. The mathematical models for the subsystems and the system are presented in a space state form (Williams II, Lawrence, 2007).
4. By using the mathematical behavioural models created by the preceding operation, it is possible to proceed to simulate system behaviour. For this purpose, the Model Predictive Control (MPC) method is used. By adjusting the parameters, the method calculates the optimal prediction output vector, the response(s) of which are presented graphically. The quality of the response obtained by using the MPC method is assessed. In this operation two decisions are possible:
   - In the case of quality behaviour simulation, we proceed with the next operation. Otherwise, changes are made in previous algorithm operations.
   - It is first determined whether or not it is possible to affect the quality of the simulating system behaviour by changing the assumptions in creating a mathematical model and changing the

parameters in the MPC method. If not, changes are made in the structural domain (operation 7). The present system elements are removed, or new system elements (subsystems or modules) are included in the system architecture. After this, all operations are repeated until the quality of system behaviour simulation is retested. This procedure may be repeated through several iterating steps until satisfactory quality of system behaviour simulation is attained. The responses obtained by such simulation may be presented graphically (by using charts) in both cases i.e. regardless of the system behaviour simulation quality.

5. Thisis followed by the system stability testing operation where the observed system and its behavioural domains (continuous and discrete) are tested by using the Lyapunov's direct method (Ljapunov, 1949).
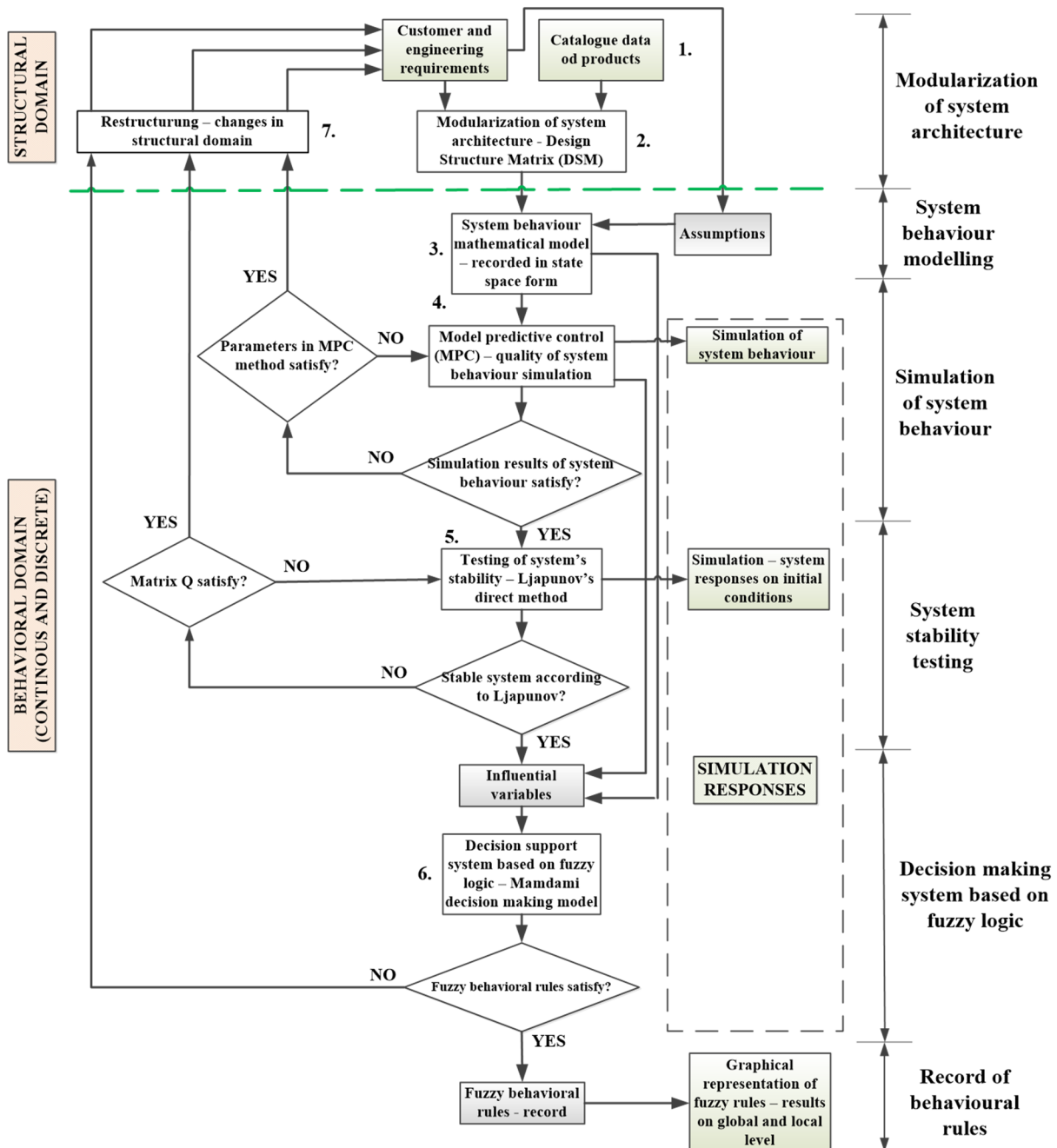

Figure 1.Complex technical system architecture development algorithm including each operation

Two cases are possible. For highly linear systems and nonlinear systems that may be presented in a linear form, system stability is tested as if the observed system were autonomous. In the case of highly nonlinear systems (that would lose their dynamic behaviour properties if presented in a linear form),

the system stability testing must take into account the system (including the MPC method). We say it needs to be observedin a controlled system. In this operation two decisions are possible:

- In case, that an unstable system is obtained according to Lyapunov, changes are made in the previous algorithm operations.
- It is first determined (depending on the system type described above) whether or not it is possible to affect system stability by changing the symmetric matrix Q. If not, changes are made in the structural domain (operation 7). Like we did with the preceding decision, we remove present or include new system elements (subsystems or modules) in the system architecture. The following algorithm operations are repeated until system stability is retested according to Lyapunov. Like we did with the preceding operation, the whole procedure may be repeated through several iterating steps until satisfactory system stability is attained (an asymptotically stable system). It is also possible to graphically present (by using charts) system's responses to the initial conditions regardless of the system stability attained.

6. Based on the influential variables obtained from the mathematical model (selected based on designer's experience) and information obtained about system behaviour (through MPC method operations and system stability testing), fuzzy rules are created (Ross, 2010). For this purpose, we use a fuzzy rule-based system (Sivanandam, Sumathi, Deepa, 2007) with Mamdamidecision-making model (Mamdami, Assilian, 1975). Based on such system behaviour rules created, the output of the decision-making system may be observed graphically, on a local or global decision (conclusion) making level. Behaviour rules may be recorded for each system variant.

7. As we mentioned before, based on the quality of the simulating system behaviour and testing of the system stability, changes are made in the structural domain. It means that the present system elements are removed, or new system elements (subsystems or modules) are included in the system architecture.

5 CASE STUDY – AIR-HANDLING UNIT

As an example to verify the proposed algorithm, an air-handling unit for air conditioning in commercial facilities (such as shopping malls, office buildings) was used. This casewas selected for the purpose of conducting the verification for an example of a highly linear complex technical system. It is provided as a combination of different functional units (technical subsystems) in accordance with the EN 1886 standard for air-handling units in buildings (1998). Case study of a complex technical system with nonlinear behaviour (load lifting device) is presented in *Osman et al.* paper (2014).

5.1. Modularization of system architecture

Based on the customer requirements received and the engineering requirements derived from them with catalogue data for similar products, the *iteration step 1*creates the initial architecture of the air-handling unit. It consists of the following subsystems: air heater, air cooler, supply and exhaust air fan, and supply and exhaust duct (as presented in Figure 2a).

The customer needs are as follows:
- whole-day uninterrupted operation of the device must be ensured,
- the device must be capable of operating in variable external conditions,
- the device must be weather resistant (resistant to water, rain, snow, ice and air),
- the device must be cable of being installed externally (on the roof of the building),
- the operation of the equipment must be silent,
- the operation of the air-handling unit must be automated,
- comfortable conditions must be provided for the device within air conditioned premises of the commercial facility.

The engineering requirements are as follows:
- the device must be capable of operating in winter and summer modes,
- the device must be capable of operating at ambient temperatures between -35 °C and 40°C,
- uninterrupted power supply must be secured (3x220 V, 50 Hz),
- the temperature of the heating medium must be 80/60 °C (heater, warm-up and pre-warm heater subsystems in air-handling unit),
- the temperature of the cooling medium must be 7/12 °C (cooler subsystem in air-handling unit),

- the temperature of incoming fresh air in winter should not exceed 25 °C and that in the summer should not exceed 19 °C,
- the temperatures within the premises should be selected in accordance with the standard applicable to commercial premises,
- it must be shock and vibration resistant,
- the lining of the air conditioner must be plasticized,
- operating stability of the air-handling unit must be ensured,
- a boiler room using natural gas to produce hot water (80/60 ºC) is envisaged for the preparation of the heating medium (hot water for the air heater),
- the air cooled condenser for cold water temperature ϑ_{hv} = 7/12 ºC for preparing the cooling medium (cold water for the cooler subsystem) is envisaged on the roof.

The iteration step 1 will start with the simplest system architecture where we can meet all engineering requirements for the premises to be used. According to the initial system architecture, the system architecture is described on a matrix basis by using a component-based DSM.

Two subsystems are added to the initial air-handling unit architecture in the *iteration step 2*: a plate heat recuperator and a steam humidifier.

In the *iteration step 3*, several subsystems were added to the preceding system architecture: a recirculation duct, a "by-pass" duct and an air warm –up heater.

In the *iteration step 4*(presented in Figure 2b), we aimed to change the previous system architecture (described in the preceding iteration step 3) by adding some new behaviour rules. For this purposes, we added a pre-warm air heater system and air preparation subsystems (zone air warm-up heaters) for three zones within the commercial facility. The matrix-based air-handling unit system architecture with potential subsystems (modules) in iteration step 4 is presented in Figure 3a. *LOOMEO* software (Tesseon, 2012) will be used for this purpose.
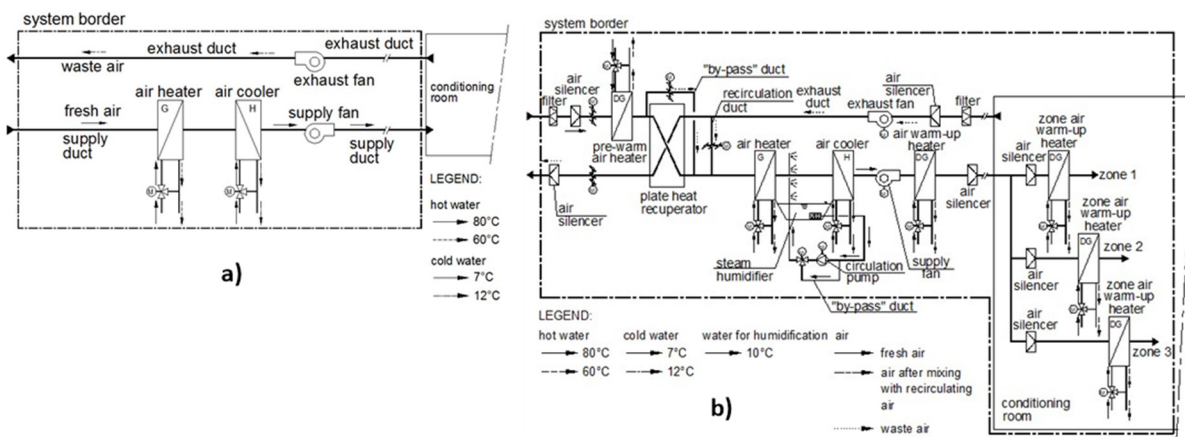

Figure 2.Air-handling unit system architecture: a) iteration step 1, b) iteration step 4

5.2. Creating the mathematical system behaviour model

The *assumptions* implemented in the modelling of each air-handling unit subsystem are as follows (*iteration step 1*):

- it is assumed that the entire system including all its subsystems is well isolated from any external impact, which means there is no external heat loss,
- the circulation of fluids (air and water) in the air-handling unit system including all its subsystems is laminar,
- although heat exchange within the heat exchanger subsystems in air-handling unit is spatial, we will assume for the purposes of describing the fluid temperature change that it depends on a single coordinate i.e. that it is one-dimensional,
- any mechanical, hydraulic or heat losses within the system are disregarded – perfect mixing of recirculated air with fresh air is assumed,
- Distribution of air in the ducts during circulation is complete and the entire system including all its subsystems is observed as systems with concentrated parameters − we thus obtain simple

models that are still capable of adequately describing the processes within the system. The process magnitudes only include the function of one variable - time t,
- the heat exchangers (recuperators) are designed as plate exchangers, but we observe them as tube-in-tube type exchangers in this paper for simplification,
- the supply and exhaust fans operate at a constant rotational speed.

When creating the mathematical model of the air-handling unit subsystem, we use the energy (heat) maintaining equation for so-called open systems (Bošnjaković, 2012). As we are using a model with concentrated parameters (for air heater and air cooler), the observed temperatures in the subsystems may be observed in terms of their mean values (Vukić, Kuljača, 2005). System of equations is present in space state model. The system log model is shownin a code and recorded in a form of file.

In the *iteration step 2*, the energy balance equation for the plate heat recuperator shows that a heat flow change occurs within the recuperator. For the steam humidifier, there is no change in energy/heat because this is adiabatic humidification (a balancing process) where h = const. (Bošnjaković, 2012). The steam humidifier circulation water pump operates at a constant number of revolutions per minute.

In the *iteration step 3*, the air warm-up heater mathematical model is the same as in the air heater (the same subsystem functionally). By adding a recirculation duct, fresh and waste air may be mixed at an appropriate ratio (air mixing system). The temperature at the exit from the mixing area and at the entry point to the air heater is thus increased, and the heat capacity of the air heater is thus reduced.

In the iteration step 4, the mathematical models for the pre-warm air heater and each zone air warm-up heater are the same as those for the air heater and air warm-up heater (executed in the previous iteration steps). The system model in state space form is presented as a code in *MATLAB* software (MathWorks, 2012a).

5.3. Simulating system behaviour by using the model predictive control (MPC) method

Based on the model created in a space state form, system behaviour is simulated by using the MPC method. The MPC method uses the influential variables, based on the selection at sole discretion of the designer who believes they have the greatest influence on the behaviour of the observed system. The influential variables are: T_V [°C] – external temperature (1), q_z [m³/s] – volume air flow (2), system variants with the following geometry: BxH [mm²] (3), and weight factor ρ [-] (4).

During the analysis of system behaviour by using the MPC method, the variables mutually vary. In addition to selecting the MPC method parameters, responses of system's output variables are simulated. System responses for the winter period presented in Figure 3b (for iteration step 4) are presented for a particular set of influential variables. The MPC method model for simulation of system behaviour was created in the MATLAB Model Predictive Control Toolbox software by using the MPC Design Tool subprogram (MathWorks, 2012b).

5.4. Testing system stability by using the Lyapunov's direct method

Stability testing is simulated for the mathematical system behaviour model in a space state form (see Section 5.2) in each iteration step by using the Lyapunov's direct method. The same influential variables (as in the preceding algorithm operation) are used. During the analysis of system behaviour by testing its stability, the variables mutually vary.

The system stability test result is presented in Figures 3c for the iteration step 4.

The testing model for system stability is created by using *MATLAB* software (MathWorks, 2012a).

5.5. Creating fuzzy behavioural rules

Based on the data obtained by using the MPC method and testing system stability, in addition to the data obtained from a mathematical model, a Mamdami fuzzy logic model is created. For this purpose, we used the decision-making system (fuzzy rule-based system).

The semantics of the influential variables used in creating fuzzy rules in iteration steps 1 and 2 are presented in Table 1, while the semantics for iteration steps 3 and 4 is provided in Table 2.

For this purpose, we used the decision-making system (fuzzy rule-based system) available in *MATLAB Fuzzy Logic Toolbox* software (MathWorks, 2012c).

Table 1. Presentation of influential variable semantics for creating fuzzy rules (in iteration steps 1 and 2)

a)

T_V [°C]	Span	Semantics
-35 – -15	0	Very low
-15 – -5	0.25	Low
-5 – 10	0.5	Medium
10 – 20	0.75	High
20 - 40	1	Very high

b)

q_z [m³/s]	Span	Semantics
0.2 – 1.25	0	Very low
1.25 – 3.6	0.25	Low
3.6 – 4.6	0.5	Medium
4.6 – 5.8	0.75	High
5.8 – 7.8	1	Very high

c)

Variant ID	Span	Semantics
1 – 2	0	Very low
2 – 4	0.25	Low
4 – 6	0.5	Medium
6 – 8	0.75	High
8 – 11	1	Very high

d)

Variant ID	$B \times H$ [mm²]	Variant ID	$B \times H$ [mm²]
1	700 x 450	7	1300 x 1100
2	1000 x 450	8	1300 x 1350
3	700 x 650	9	1600 x 1350
4	700 x 750	10	1600 x 1700
5	1000 x 750	11	1950 x 1700
6	1000 x 1100		

e)

ρ [-]	Span	Semantics
0 – 0.2	0	Very low
0.2 – 0.4	0.25	Low
0.4 – 0.6	0.5	Medium
0.6 – 0.8	0.75	High
0.8 – 1	1	Very high

5.6. Record of fuzzy behavioural rules and their graphical representation

The system's fuzzy behavioural rules will be registered in the form of file. The results of global conclusions for iteration step 4 are presented in the form of the results obtained in Figure 3d. For this purpose, we used *MATLAB Fuzzy Logic Toolbox* software to record fuzzy behavioural rules in the form of a *fis* file.

Table 2. Presentation of influential variable semantics for creating fuzzy rules (in iteration steps 3 and 4)

a)

T_V [°C]	Span	Semantics
-35 – -15	0	Very low
-15 – -5	0.25	Low
-5 – 10	0.5	Medium
10 – 20	0.75	High
20 - 40	1	Very high

b)

q_z [m³/s]	Span	Semantics
0.2 – 1.25	0	Very low
1.25 – 3.6	0.25	Low
3.6 – 7.8	0.5	Medium
7.8 – 13.9	0.75	High
13.9 – 27.8	1	Very high

c)

Variant ID	Span	Semantics
1 – 3	0	Very low
3 – 6	0.25	Low
6 – 9	0.5	Medium
9 – 12	0.75	High
12 – 16	1	Very high

d)

Variant ID	$B \times H$ [mm²]	Variant ID	$B \times H$ [mm²]
1	700 x 450	9	1600 x 1350
2	1000 x 450	10	1600 x 1700
3	700 x 650	11	1950 x 1700
4	700 x 750	12	1950 x 2000
5	1000 x 750	13	2200 x 2000
6	1000 x 1100	14	2200 x 2300
7	1300 x 1100	15	2500 x 2300
8	1300 x 1350	16	2500 x 2600

e)

ρ [-]	Span	Semantics
0 – 0.2	0	Very low
0.2 – 0.4	0.25	Low
0.4 – 0.6	0.5	Medium
0.6 – 0.8	0.75	High
0.8 – 1	1	Very high

5.7. Presentation of the obtained results

Figure 3 present the obtained results for the air-handling unit architecture in the winter period, for iteration step 4. A more detailed discussion of the results will be provided in Section 6.

6 DISCUSSION OF THE RESULTS

As regards the example of a complex air-handling unit technical system, only the winter period operation was verified because it was the only time it made sense to study the influence of uncertain operating conditions. Uncertain operating conditions result from low external temperatures that mostly affect the stability of the technical system. The system is highly linear and was as such observed as part of the research. It may be mathematically described by using a set of ordinary differential equations without using a prior linearization procedure. As the air-handling unit is a thermo-hydraulic system of a predominantly continuous nature, it subsystems was tested in the continuous domain. The discrete domains (fan electric motors, three-way valve motors and damper motors) are not contemplated in this research because their influence on system operation is not significant, especially under uncertain operating conditions.

By simulating technical system behaviour and testing its stability (by reviewing the fuzzy rules and the results of local and global decision model concluding in each iteration step), several facts are observed:

- external temperature T_V ranging between -35 and -25 °C have the greatest influence on system stability as a result of a very high heat transfer coefficient α at such low air temperature,
- the second greatest influence on system stability have high circulation speeds, and high volume air flows q_z, with smaller air-handling unit geometries (by this we mean the surface area of the *BxH* chamber cross section).
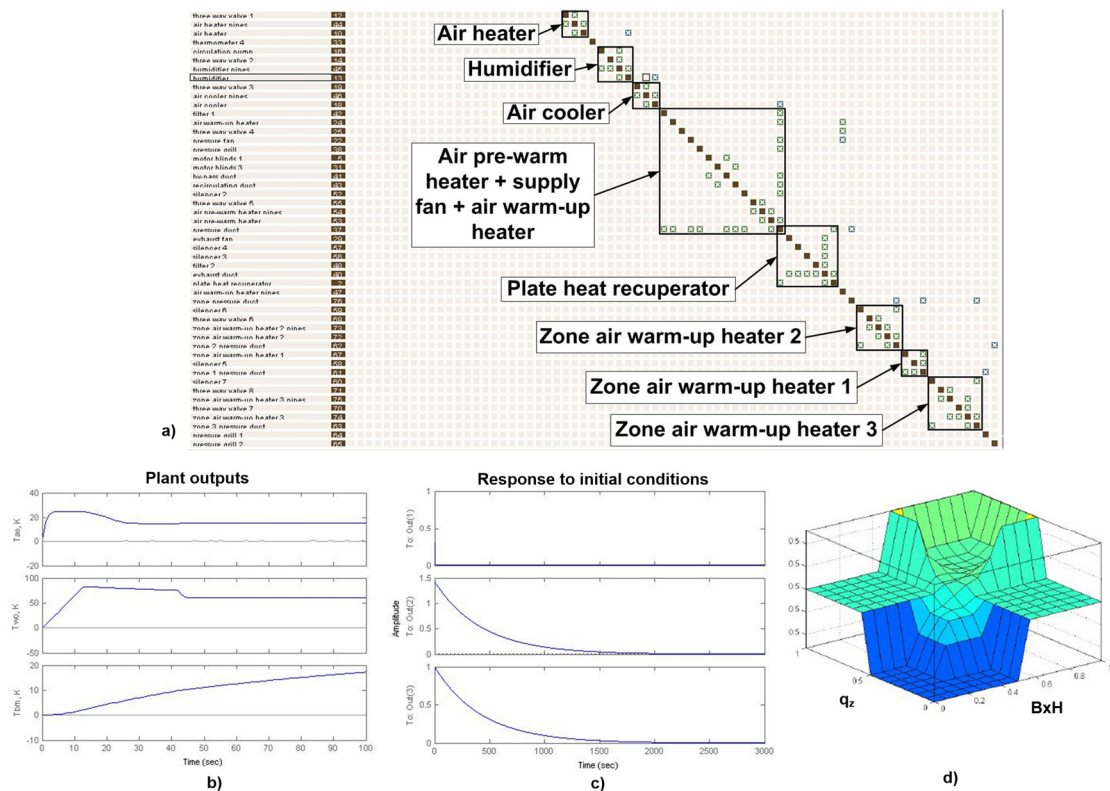

Figure 3. Results obtained for the air-handling unit system architecture in the winter period (iteration step 4); a) component-based DSM, b) MPC method – output variable responses, c) stability testing by using the Lyapunov's direct method, d) results of global decision model concluding for input variables q_z and BxH

By adding subsystems (pre-warm air heater, plate heat recuperator, air mixing unit, air warm-up heater) to the air-handling unit during the respective iteration steps, a much higher temperature was achieved at the entry point to the last subsystem. A large temperature difference we had in one subsystem was divided among several of them by introducing new subsystems. This also reduced the capacities of the other subsystems and ensured stable operation of the technical system. Iteration step 4 demonstrated that it was possible to use this approach by introducing new behavioural rules with satisfactory results.

7   CONCLUSION AND FUTURE RESEARCH

The algorithm presented in this paper was found to have a good chance of being developed into an efficient and quick tool and an aid to designers. It would be used in the development of new and improving existing system architectures in early stages of the design processes. Bilateral mapping of data between the structural model and behavioural model in complex technical systems allows us to record system behaviour both under normal operating conditions and in uncertain situations. The algorithm also allows us to include disturbance variables in the mathematical behavioural model, which cause uncertain operating conditions. Iterative steps in the algorithm enable us to reach satisfactory system architecture relatively quickly through changes in the structural and behavioural domain. The time required for the implementation of the algorithm depends primarily on the system complexity, as well as the capabilities of computer to carry out complex mathematical calculations.

After evaluating the test results in the development of technical systems, the algorithm was found to be very suitable for systems with very slow processes i.e. processes with time delays (such as the air-

handling unit presented in this paper).This also enhanced the robustness of the system architecture and its quality, thus making it immune to potential changes caused by uncertain operating conditions. In addition, the proposed algorithm enables simplifying, improving and accelerating product development processes, primarily those exposed to uncertain operating conditions.

The proposed algorithm should be extended to enable monitoring changes for each subsystem component, that is, to determine which subsystem component should be included in or removed from the subsystem architecture to achieve its satisfactory behaviour and stable operation in uncertain operating conditions. Transfer of information between operations in presented algorithm and feedbacks about expected behaviour at this stage is still ad-hoc. Also, one of the possible directions of future research could be automated transfer of information between algorithm operations. Development of a computer programm with implemented algorithm presented here could greatly facilitate the application of them for someone who is not an expert in particular methods in algorithm operations.

REFERENCES

(2012) …*LOOMEOSoftware*, Documentation, Tesseon GmbH, Germany

(2012a) …*MATLAB, User's guide*, documentation, MathWorks

(2012b) …*MATLAB Model Predictive Control Toolbox, User's guide*, documentation, MathWorks

(2012c) …*MATLAB Fuzzy Logic Toolbox, User's guide*, documentation, MathWorks

…EN 1886 – Ventilation for buildings – Air Handling Units – Mechanical Performance, 1998.

Bošnjaković, F.(2012)Thermodynamics I, II, III (in Croatian). Graphis, Zagreb, Croatia

Buss, M., Glocker, M., Hardt, M., von Stryk, O., Bulirsch, R. and Schmidt, G. (2002) Nonlinear Hybrid Dynamical Systems: Modelling, Optimal Control, and Applications, In: Analysis and Design of Hybrid Systems - Lecture Notes in Control and Information Science (LNCIS), Springer Berlin, pp. 311–335.

Camacho, E.F. and Bordons, C. (2004) Model Predictive Control. Springer – New York, USA

Diepold, K.J., Biedermann, W., Eben, K.G.M., Kortler, S., Lohmann, B. and Lindemann, U. (2010) Combining Structural Complexity Management and Hybrid Dynamical System Modelling. In: 11th International Design Conference DESIGN 2010, Vol. 2, Dubrovnik, Croatia, pp. 1045 – 1054.

Diepold, K.J., Winkler, F.J. and Lohmann, B. (2010) Systematical hybrid state modelling of complex dynamical systems: The quad - I/HS framework. Mathematical and Computer Modelling of Dynamical Systems, Vol. 16, No. 4, pp. 347 – 371.

Du, X. and Chen, W. (2000) Methodology for Managing the Effect of Uncertainty in Simulation-Based Systems Design.AIAA Journal, Vol. 38, No. 8, pp. 1471-1478.

Du, X. and Chen, W. (2001) Efficient Uncertainty Analysis Methods for Multidisciplinary Robust Design.AIAA Journal, Vol. 40, No. 3, pp. 545-552.

Kota, S. and Chakabarti, A. (2009) A Method for Evaluation of Product Lifecycle Alternatives under Uncertainty. In: 17th International Conference on Engineering Design ICED '09, pp. 289-300.

Ljapunov, A.M. (1949) The General Problem of the Stability of Motion.Ph.D. thesis, Ann. Math. Studies, Princeton University Press, New Jersey, USA

Mamdami, E.H. and Assilian, S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-machine Studies, Vol. 7, No. 1, pp. 1-13.

Osman, K., Štorga, M. and Marjanović, D. (2014) An Approach to Nonlinear Behaviour Prediction of Technical Systems, In: 13th International Design Conference DESIGN 2014, Vol. 2, Dubrovnik, Croatia, pp. 1307-1318.

Osman, K., Štorga, M., Stanković T. and Marjanović, D. (2011) Behaviour Prediction Framework in System Architecture Development, Applied Mechanics and Materials, Vol. 104, No. 3, pp. 3-12.

Pimmler, T. U. and Eppinger, S. D. (1994) Integration Analysis of Product Decompositions. In: 1994 ASME Design, Theory and Methodology Conference, Minneapolis, USA

Ross, T.J. (2010) Fuzzy Logic with Engineering Applications. third edition, John Wiley & Sons, Ltd., USA

Sivanandam, S.N., Sumathi, S. and Deepa, S.N. (2007) Introduction to Fuzzy Logic using MATLAB. Springer – Verlag, Berlin Heidelberg, Germany

Steward, D. V. (1981) The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transactions on Engineering Management, Vol. 28, pp. 71-74.

Vukić, Z. and Kuljača, Lj. (2005) Automatic Control – Analysis of Linear Systems (in Croatian). Kigen, Zagreb, Croatia

Williams II, R. L. and Lawrence, D. A. (2007) Linear State – Space Control Systems. John Wiley & Sons, Ltd., New Jersey, USA