Qualitative Monitoring of the Consequences of AI Solutions in Safety-Critical Systems

Mark Tappe,[1] **Benjamin Kelm,** [2] **Oliver Niggemann,** [1] **Stephan Myschik** [2]

[1] Institute of Automation Technology, Helmut Schmidt University, Hamburg, Germany
[2] Institute for Aeronautical Engineering, University of the Bundeswehr, Munich, Germany
mark.tappe@hsu-hh.de,

Abstract

Today, Cyber-Physical Systems (CPS) are often used in safety-critical situations. More and more, Artificial Intelligence (AI) and especially data-based methods, i.e. Machine Learning (ML), are used to increase the adaptability of systems. This immediately leads to a security risk, since data-based methods usually learn a black-box model (e.g. neural network or reinforcement learning). To still use these AI methods for safety-critical systems, like anomaly detection, optimisation or reconfiguration tasks, a supervision tool is needed.

In order to enable safe operation of data-based ML algorithms and to make statements about the stability of the system we present an implementation of qualitative monitoring of the system behaviour in the context of reconfiguration. This leads to the next problem, as a qualitative state prediction tends to branch infinitely for complex systems. Our approach limits the state prediction to the states with immediate impact. To achieve this goal and to visualise the effects for a supervision task a virtual structure similar to decision trees is implemented to generate an overview of the upcoming predicted system states. In addition, the behaviour of the system variables is extracted from the qualitative states in order to determine the risk of a predicted state.

In summary, this algorithm acts as an independent supervision agent for various AI/ML algorithms and alerts when risks are detected during operation. We can show that different reconfiguration options for a CPS with abnormal behaviour can be successfully evaluated in order to transfer the CPS as safely as possible to a new state.

1 Introduction

Cyber-Physical Systems (CPS) are very prevalent in our modern times, as the integration of microcomputers and other advanced technology offers a significant impact for a systems computational and communicative capabilities, see (Baheti 2011) and (Wolf 2009). To improve their performance a high level of technical expertise is required, which is often associated with high costs. Therefore, Machine Learning (ML) algorithms are often used for optimisation tasks based on existing data sets. However, once AI and data-based modelling determine the way a system operates, we lose predictability. This issue is of utmost importance because AI solutions, particularly data-based methods like many ML algorithms, typically create black box models based on given data (Tjoa and Guan 2021) and (Wan et al. 2021). When a system's behaviour is solely determined by measurement data, it cannot be fully defined. In safety-critical applications, this poses significant risks as infallibility cannot be verified. Although some solutions based on ML approaches, such as the Safety+AI approach of (Gheraibia et al. 2019), have been researched, we aim to focus on qualitative reasoning instead.

> "Reasoning about, and solving problems in, the physical world is one of the most fundamental capabilities of human intelligence and a fundamental subject for AI."

These words of Bredeweg (Bredeweg 2003) show very well our motivation for our approach. Our goal is to design a supervision agent that is able to monitor the behaviour of a system and to estimate the consequences of AI interventions. In theory an extensive numerical simulation would be able to evaluate those consequences very accurate, but especially for CPS, which combine computational science with engineering disciples this is not a trivial task and such a simulation is often not available. For this purpose, we investigate the possibility of using qualitative system models, based on a general system description, instead of complex simulations.

The benefits of such a prediction approach are examined in the joint project (K)ISS[1]. The aim of the project is to monitor the safety-critical life support system of the ISS module COLUMBUS. We aim to reconfigure the system by activating redundant components based on detected faults, ensuring effective recovery. To validate the reconfiguration process and assess different AI decisions, we successfully implement our approach for a supervision agent.

In section 2 we will explore general concepts related to qualitative system representations, and then in section 3 we will present our solution based on qualitative reasoning. The application of this will be in section 4 using a simulated environment. Finally, we will conclude this work in section 5 and provide an outlook on future tasks and challenges.

[1](K)ISS is part of dtec.bw®, see Acknowledgements for funding information
https://dtecbw.de/home/forschung/hsu/projekt-kiss

2 State of the Art

Before our solution is presented in section 3, we will first present a general overview of current approaches and explain their shortcomings, which we encountered during our research.

Safety Analysis of AI and the Shortcomings of Data-Based Models

As long as system measurement data is available the behaviour of a CPS can be learned. A basic application is to learn and formulate this behaviour in form of a timed automaton. As an example for how a automaton can be learned, we look at the algorithm of HyBUTLA, presented by (Niggemann et al. 2021). This algorithm constructs a timed automaton, which can be learned from system measurements, to describe the behaviour of a system, see Figure 1. In general the steps to learn the behaviour of such a system can be described with:

0: Record and synchronize the signals of the CPS.

1: Generate a list of discrete events.

2: Construct a tree based on the recorded events.

3: Simplify the tree by merging similar nodes.

The BUTLA algorithm, which depends on positive data examples, still has shortcomings. In certain cases, anomalies can occur that are not identified or are incorrectly identified. This happens because the data of error cases is not available and therefore there is a deficit of information.

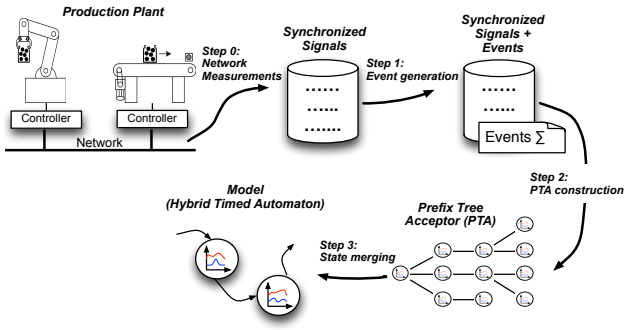

Figure 1: General concept of learning a timed automaton, the steps refer to the HyBUTLA algorithm, see (Niggemann et al. 2021)

Other data-based solutions, like many different ML algorithms, are commonly used, but their black-box nature hinders understanding and verification, especially of their internal workings - see (Tjoa and Guan 2020). The idea to translate and encode the behaviour of those AI models is part of the research of Explainable Artificial Intelligence (XAI). But most approaches of XAI are not universally valid. Instead, methods like saliency maps help with specific use cases like image recognition, but are more difficult to apply to decision process models. They can be used, but even if a correlation between input and output can be established, the result is by far not sufficiently precise enough to determine the internal decision process. Conventional algorithms such as decision trees, which do not operate on the same basis as AI algorithms, are much more reliable. But as shown in the work of (Wan et al. 2020), they lack in performance. They have shown the accuracy of decision trees in comparison to neural networks in image detection is behind by up to 40%.

Our approach emphasises the importance of decision tree reliability and combines it with qualitative reasoning for the safety assessment algorithm.

Qualitative Reasoning based on QSIM

In contrast to data based models, we can describe a system instead by its qualitative behaviour. The qualitative behaviour of a system is based on available system knowledge, which also grants information about non-measurable states. A promising concept about qualitative description of system behaviour was presented by (Kuipers 1986) in the papers about the QSIM algorithm. The used notation has been recognised by various scientists (Simon 1991; Say and Kuru 1996; Trave-Massuyes, Ironi, and Dague 2003), which is why we will also use its notation in this paper to describe a qualitative system.

The theory behind QSIM is to mimic the differential equations of classical systems with qualitative differential equations (QDE). A QDE would describe how a qualitative state can change. For each parameter P (which is basically a system variable) the qualitative state QS would be defined at a qualitative point in time t_i in the form of a tuple consisting of a discrete qualitative value and the direction of change. An example is given with:

$$QS(P, t_i) = \langle val, dir \rangle \tag{1}$$

The discrete value val can be defined as a single point value or as a pair of values specifying an interval in which the current qualitative value lies. In order to capture the change of a state, it is assigned an additional direction of change dir, which can take one of three variants: steady, increasing or decreasing. In addition, there is a discrete range of values for each parameter called the quantity space, which contains all known discrete values of that parameter - known as landmark values. To include multiple qualitative states in this kind of formulation a set F containing multiple parameters $F = \{P_i, ...\}$ can be created. Based on this a whole system can be defined by $QS(F, t_i)$.

To represent the QDE, which define the qualitative behaviour of a system, a set of constraints is needed, each limiting the possible transitions of the qualitative states opf the parameters. A comprehensive list is shown in Table 1. To depict a more complex ordinary differential equation, the equation can be separated in multiple elementary functions, which can then be translated in qualitative constraints. In some cases a constraint might change if the system reaches or leaves a set operating point. This can be handled by defining restrictions, which define which constraints apply for a given set point. A geometric function such as the sine can represent its cyclic effect with restrictions and alternating between M+ and M- constraints.

ADD(X, Y, Z)	$Z(t) = X(t) + Y(t)$
DERIV(X, Y)	$dX/dt = Y(t)$
M+(X, Y)	$X(t) = f(Y(t))$, where $f' > 0$
M-(X, Y)	$X(t) = f(Y(t))$, where $f' < 0$
MINUS(X, Y)	$X(t) = -Y(t)$
MULT(X, Y)	$Z(t) = X(t) * Y(t)$
CONST(X)	$X(t) =$ constant value

Table 1: List of Qualitative Constraints, complemented version of (Say and Kuru 1996)

Qualitative reasoning, similar to QSIM, has been researched and developed in the field of discrete model diagnosis. These approaches are often specific to certain toolboxes and proprietary applications, see (Williams et al. 2003; Struss and Price 2003). The fundamentals are widely known and were the focus of multiple research papers (de Kleer and Brown 1984; Dvorak and Kuipers 1989; de Kleer 1993), but since the year 2000 the application of qualitative simulation shifted. The numerical simulations became more reliable thanks to the increased computing power of computers, and the qualitative analyses were used more for the theoretical discussion of abstract systems and interrelationships, such as the effects on the population of species in (Salles and Bredeweg 2006).

In (Bredeweg 2003) the main issues and some open tasks of qualitative simulation back in 2003 were highlighted, particularly the modality of qualitative systems. On the one side this modality allows users to create diverse model libraries, which can be reused in different ways, but on the other hand each qualitative analysis needs a different degree of abstraction and detail and a uniform system did not exist back then. This problem continued with a lack of integration in standard engineering and research tools. In (Klenk et al. 2014) this problem got tackled by combining the usage of Modelica models with the ideas of qualitative reasoning. They achieved the goal to generate the qualitative model mapped upon existing modelica models, which negates the need for an additional modelling step. On the other hand we transferred the principles of QSIM and QDEs into the modern programming language python, which is especially well used in the machine learning community as another implementation. In this paper we will not further expand on the topic of implementation, but instead focus on the concept how this qualitative description can be used to evaluate the behaviour of a system. Still we are taking a custom take on the implementation to focus the constraints more on system dependencies instead of ideal QDEs.

Identification of Anomalies and Faults

For the sake of completeness, the need for identification and diagnosis of errors should be noted. One may assume that a failure is feasible via the QDEs defined above, but their algorithm, depending on implementation, cannot deduce the source of a defect. Still the underlying fundamentals of neglecting a specific mathematical model can be applied as well.

The work of (de Kleer and Williams 1987) shows how the shift of model-based diagnosis shifted from specific fault models towards the tracking of an inconsistent behaviour as indicator of a fault. Based on this, there are various alternative methods for detecting anomalies and faults that do not even require a mathematical model, as CPS usually provide a comprehensive database. These data-based algorithms can be evaluated as multi-time variant data sets and serve as a basis to describe the system behaviour of the plants from observations. Based on this data, it is then possible to create data-based models such as the Univariate Fully-Connected AutoEncoder (UAE), whose good performance was described by Garg et al. (Garg et al. 2022). However, their limitations were also pointed out, as these solutions are often limited to a specific use case, for example the UAE's performance decreased when used for a system with multiple operating states.

Still those algorithms perform well and there is no need to apply an additional supervision layer on top. In the later context, we assume that the identification of an anomaly and the diagnosis of faulty system components is available as a basis for the reconfiguration task.

3 Solution

In this section, we address implementing a qualitative monitoring agent for a CPS. We'll explain the generated input during reconfiguration, the use of QSIM basics in our supervision agent, and risk evaluation for predicted states. This guides the selection of a reconfiguration option with the lowest expected risk.

Assumptions for this paper: The system is faulty, but the cause is diagnosable and faulty components got detected. We aim to find a reconfiguration that adjusts the system structure to return to a safe workspace.

Reconfiguration

Generally, the goal of the supervision layer is to identify those possible configurations of the system that yield a safe and stable system. The actual identification of possible, valid configurations is typically performed by a reconfiguration program. Here, we would like to present the implemented reconfiguration algorithm *AutoConf* in brief, which is detailed and applied to ECLSS by (Kelm et al. 2022).

AutoConf, a qualitative model-based reconfiguration algorithm using Satisfiability Theory (SAT), was recently presented by Balzereit and Niggemann (Balzereit and Niggemann 2022). It can be used for the reconfiguration of hybrid systems and is divided into two main steps. In the first step a logical formula which represents the reconfiguration problem is created. In the second step, this formula is solved by a SAT solver.

The first step in creating the logical formula, known as the qualitative system model (QSM), involves generating causal graphs G that define the relationships between inputs and system states, e.g. a qualitative description of system dynamics. The inputs, represented as binary values (e.g., valve opened or closed), are denoted as $B = \boldsymbol{b}_1, ..., \boldsymbol{b}_k$. The causal graph is divided into positive ($G^+ = (V, E^+)$) and negative ($G^- = (V, E^-)$) subgraphs, indicating their influence on

state variables. The nodes in the graphs include states and inputs ($V = \boldsymbol{x}_1, ..., \boldsymbol{x}_n, \boldsymbol{b}_1, ..., \boldsymbol{b}_k$), while the edges in the positive graph E^+ represent significant state increases when inputs are activated. The negative graph E^- represents significant state decreases.

Next, the algorithm encodes the causality into propositional logic by using symbols (low_{x_i} and $high_{x_i}$) to represent state limits. These symbols indicate whether a state is below the lower limit or above the upper limit and, consequently, imply the activation or deactivation of certain inputs. Binary logical connectives (implication [$\Rightarrow$], negation [$\neg$], conjunction [$\wedge$], and disjunction [$\vee$]) are used to formulate constraints. For instance, if a reservoir exceeds its limit, the formula implies either opening an outflow or closing an inflow.

In the second step, a logical SAT solver is employed to solve the logical formula, utilising logical reasoning. If the formula is satisfiable, it means there exists an assignment of input variables that satisfies the formula. This assignment corresponds to the new configuration required to achieve a valid system state within a specified reconfiguration time Δt_{rcfg}. If the formula is not satisfiable, a reconfiguration is not possible, and the system may need to be shut down.

Generally there are multiple valid configurations that are solutions of the reconfiguration problem, which can be iterative listed by negating the previously found solution and searching for another solution. To identify the best solution, e.g. the solution with the lowest risk of instability, a supervision layer is required.

Qualitative Supervision

To assure a safe operation of safety-critical systems during after a detected fault, the reconfiguration needs to be evaluated to prevent malfunctions. Therefore we want to design an supervision agent, to monitor the qualitative consequences of such actions.

Previously we presented the QSIM algorithm by (Kuipers 1986) and described how it can be used to abstract the behaviour of a system. In contrast to QSIM we added the F+ and F- functions. These function behave similarly to the M+ and M- functions in the original, but additionally investigate the dependencies of 0-values. The added F+ and F- functions are not monotonously increasing or decreasing functions, but allow a saddle point behaviour at a discrete value of $\langle 0 \rangle$. This is due to the fact that the dependencies on the input configuration represent a dependency on binary values, which can be implemented more efficiently by allowing a steady 0-value. In the case a system component is not needed and therefore shutoff, the function can be deactivated and then take on the classical M+ or M- behaviour once the component is reactivated.

The qualitative variables are initialised at t_0 in the form of:

$$\begin{aligned} QS(P_i, t_0) &= \langle val, dir \rangle \\ val &\in [0, \textit{too low}, \textit{low}, \textit{norm}, \textit{high}, \textit{too high}, +\infty] \\ dir &\in [\textit{dec}, \textit{std}, \textit{inc}] \end{aligned} \tag{2}$$

The qualitative values val of those variables are discretised measured values, which are categorised as low, $norm$ or $high$ depending on the known limits of their working range or *too high* and *too low* if the boundaries are exceeded. Additionally, their current change of direction is depicted with dir - increasing, steady or decreasing.

Combining qualitative findings with reliable system representation allows us to use a decision tree structure to understand system behaviour. We introduce the Qualitative Analysis Tree (QuAT) for this purpose. A simplified example is illustrated in Figure 2. Starting from an initial qualitative state 0, we assess its constraints to find possible transitions (e.g., a and b). As transitions occur, new qualitative states emerge, and their constraints are evaluated for predecessor states. If a transition leads to a steady state or detects a risk (e.g., transition b), further evaluations cease. The topic of risk assignment will be covered in the upcoming subsection. Nonetheless, to predict the comprehensive system state, we also consider subsequent states of successors, as they might appear deceptively safe, as seen in Figure 2 ($0 \Rightarrow a \Rightarrow 1 \Rightarrow d \Rightarrow Risk$).

If each successor state is evaluated we would obtain a qualitative description of the entire system like the original QSIM application. However, this approach becomes incredibly complex due to its combinatorial nature. To address this challenge, we reduce the number of iterations for our qualitative evaluation. Predicting the behaviour over a short abstract time horizon can still be highly effective, as each discrete qualitative time-step represents a specific event or a significant change of parameter values. Long-term analysis often isn't necessary as short-term defects have more serious consequences that require immediate prevention. Any negative long-term effects can be corrected with ongoing reconfiguration inputs.

The supervision agent's goal isn't finding optimal transitions but spotting safety-critical states after transitions. At a minimum, the next states, including all possible transitions, are analysed.

Validation of Analyzed States

Once the system's behaviour can be qualitatively analysed, it allows for evaluating its behaviour as a predictive model for future steps. By performing the qualitative algorithm for each discrete event, the upcoming behaviour can be analysed. As mentioned before we can create a QuAT whose tree structure consisting of successor states allows us to determine the qualitative system behaviour in the next discrete time points. Valid transitions can be assigned a positive score based on the operating range of each parameter, indicating that those states are considered acceptable.

But how is this score defined? The operating range for each variable is known and therefore we can estimate if a qualitative value becomes *too high* or *too low*. If these parameters exceed predefined safety limits, they are identified as risky states. Predicted states, which direction of change is not $steady$ pose a minor risk as they, potentially lead to limit violations later on. By combining the evaluation of the qualitative values and their direction of change, a risk score can be estimated and assigned to each state.

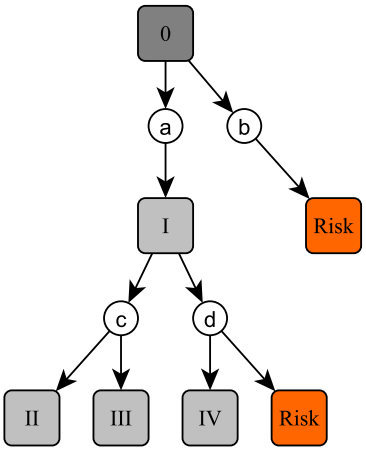

Figure 2: Exemplary QuAT for the representation of how supervision of system states is carried out. Each positional-state is represented with a box and roman numerals, while the possible transitional-states are marked with circles and small letters. States which oppose a risk are shown in orange.

Algorithm 1 Safety assessment based on qualitative risk assignment

Input: Current data of the system
Model: Qualitative system description, based on set F
Output: Risk-analysis of transitions

1: Discretize input data.
2: Initial qualitative state $QS(F, t_0)$ is set as $QS(active)$.
3: **while** qualitative prediction **do**
4: Analyze successor states of $QS(active)$.
5: Add all valid states to $ActiveList$.
6: remove $QS(active)$ from $ActiveList$
7: set next state from $ActiveList$ to $QS(active)$
8: **end while**
9: $\Rightarrow$ create $Tree$, with nodes of all qualitative states
10: **for** each $state$ in $Tree$ in bottom-up order **do**
11: **if** $state$ is leaf node **then**
12: Assign estimated risk
13: **else**
14: Update risk, based on successor nodes
15: **end if**
16: **end for**
17: **return** risk and qualitative behaviour

If further insights into the system are available and the risks associated with the interaction of specific parameters are known, these effects can be easily detected based on the qualitative state descriptions. Interdisciplinary effects should be considered when creating the qualitative state constraints. Similar to backpropagation in a neural network, risk estimations can be applied to predecessor states, enabling the assignment of a validity score to the entire QuAT, as during the creation of a QuAT the intermediate states can't be fully evaluated without knowledge about how their child states behaviour.

Algorithm 1 demonstrates an implementation example. Predefined qualitative system descriptions and system measurements are essential to define the initial qualitative state (Line 1-2). This includes the assignment of measurements to known qualitative discrete quantities, but also the representation of the system intervention that is to be studied. The current state undergoes qualitative simulation (Line 3-8) until the prediction horizon is reached or a steady state is attained. Analysed states are then organised into a tree structure, illustrating the system behaviour (Line 9). The risk assignment (Line 10-16) follows two main steps: Firstly, the tree is evaluated in a bottom-up manner, starting with the risk estimation of the leaf nodes. Afterwards their predecessors are updated primarily by their successors' risk. Once evaluation of all qualitative states is completed, the output contains the risk analysis of the current state's transition (Line 17).

The output of the safety assessment can depend on the use-case. One option would be to return the estimated risk for the current possible transitions, to validate if a specific transition should be avoided. Alternatively, the whole tree with the updated risk scores can be returned to present the system engineers a current overview of the system and its upcoming behaviour. The latter case is particularly important in situations where multiple safety-critical states are identified, requiring operators to navigate the system during challenging operations.

4 Application in Safety Assessment and Supervision of AI Solutions

This section covers the application of the monitoring agent for a CPS, here the COLUMBUS module of the ISS. The knowledge about upcoming system states, especially in terms of the assessed risk, is essential for a safe and secure operation.

CPS System Description - ISS Columbus ECLSS

The COLUMBUS module is the biggest contribution of the European Space Agency (ESA) to the International Space Station. Its purpose is to serve as a unique platform for different fields of research: Human physiology, biology, fundamental physics, material sciences and fluid physics. Furthermore, external experiment facilities allow the long-term and non-perturbed observation of the Earth and the universe. The European laboratory is operated by the COLUMBUS Control Center at the German Space Operations Center nearby Munich (Doyé 2012).

The most critical and vital system of the COLUMBUS module is the Environmental Control and Life Support System (ECLSS), whose topology is shown the process flow diagram in figure 3. It consists of a supply (ISFA) and return (IRFA) fan assembly, a redundant pair of cabin fan assemblies (CFA 1/2), a temperature control valve (TCV), which distributes the airflow into two redundant cooling and condensation cores (Core 1 and 2) within the condensate heat exchanger (CHX) to cool and dehumidify the air.

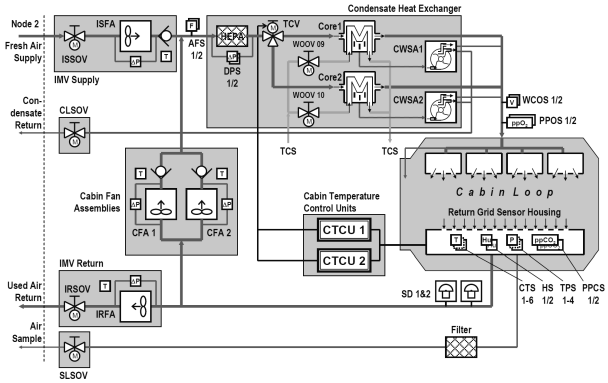

Figure 3: Cabin Loop of ISS ECLS-System by (Doyé 2012)

The airflow is then channelled into the cabin, where it mixes with the cabin air. To refresh the air and ensure smoke detection, a minimum volumetric flow rate has to be passed by the smoke detectors (SD 1/2) and is returned by the ISFA and recycled in part through the CFAs. The thermal control system (TCS) is composed of the Cores, the coolant and external heat exchangers and is controlled by the redundant cabin temperature control units (CTCU 1/2).

Additionally, there are multiple sensors, measuring the volumetric airflow (AFS), pressure differentials across fans and filter (ΔP or DPS), partial pressure of O_2 and of CO_2 gas (PPOS/PPCS), cabin temperature (CTS 1-6), humidity (HS 1/2) and the total pressure (TPS 1-4).

Reconfiguration of a Fault Case

Consider the following hypothetical failure case for illustration purposes: An accident occurs in the COLUMBUS module during an experiment, resulting in the failure of Cooling Core 1. The cabin's pressure has increased beyond the threshold due to gas leakage, and the hatch has been closed after the accident. The initial system state before reconfiguration is represented by

$$\begin{aligned}\boldsymbol{x}^0 &= [T_c, \phi_c, \dot{V}_{AFS}, p_c]^T \\ &= [303\,\mathrm{K}, 0.50, 500\,\mathrm{m}^3/\mathrm{h}, 103.5 \times 10^3\,\mathrm{Pa}]^T\end{aligned}$$

and the input configuration by

$$\begin{aligned}\boldsymbol{b}^0 = [&b_{\mathrm{ISFA}}, b_{\mathrm{IRFA}}, b_{\mathrm{CFA}_1}, b_{\mathrm{CFA}_2}, ... \\ &b_{\mathrm{TCV}_1}, b_{\mathrm{TCV}_2}, b_{\mathrm{C}_1}, b_{\mathrm{C}_2}]^T \\ = [&1,0,0,1,1,0,1,0]^T.\end{aligned} \tag{3}$$

We thus have only ISFA, CFA2 and one cooling branch (TCV1, C1) activated, which corresponds to the default configuration, where the used air is returned over the hatch opening.

We also find, by an underlying fault diagnosis algorithm, that two actuators have failed. The health state is given by

$$\boldsymbol{h}^0 = [1,1,1,0,1,1,0,1]^T. \tag{4}$$

Using the causal graph, the reconfiguration algorithm classifies the inputs into inflows and outflows. These are then transformed into a logical set of formulas using *AutoConf ext*. The formulas aim to answer the question:

> Which inputs do I need to open or close to bring the corresponding state within acceptable bounds?

An excerpt of the logical formula demonstrates the implications of a high temperature, where either one of the cooling cores (b_7 or b_8) or the ISFA fan (b_1) need to be activated:

It shows the implications of a high temperature, which are to switch on either one of the cooling cores (b_7 or b_8) or to switch on the ISFA fan. The negation of the pre-reconfigured inputs (b^0) excludes inputs that are already reconfigured. Actuator dependencies and internal flow structures are also included in the logical formula.

The logical formula is then checked for satisfiability using Z3. If it is satisfiable, a model (input assignment) that satisfies the formula can be obtained. In this fault case, the formula is satisfiable, and the algorithm proposes a new input configuration to recover the system:

$$\boldsymbol{b} = [1,1,0,0,0,1,0,1]^T. \tag{5}$$

By activating the ISFA and the second cooling branch (TCV2 and C2), the pressure can be reduced, and the temperature can be lowered. Note that there exist multiple valid configurations (e.g. CFA1 could also be switched on). If the logical formula is not satisfiable, the system is shut down. Alternatively, constraints can be relaxed to lower the system requirements and prioritize certain state variables. The output is then presented as a list of possible configurations that solve the logical formula, the supervision agent will then select the safest system intervention.

Supervision of the Reconfiguration

The reconfiguration evaluates the current system state and determines a possible system configuration on the basis of the system's stability status, which is intended to bring the system to a stable state in the event of an anomaly. This procedure was described before using the system pressure and the cabin temperature as examples. As long as the logical formula can be solved with the *AutoConf ext* algorithm, several alternative configurations in the form of equation 6 can usually be determined. All of them can remedy the anomaly that has occurred as they solve the logical formula presented in the reconfiguration approach.

$$\begin{aligned}\boldsymbol{b}_0 &= [10001000]^T \\ \boldsymbol{b}_1 &= [11001000]^T \\ \boldsymbol{b}_2 &= [01101000]^T \\ \boldsymbol{b}_3 &= [00100100]^T \\ \boldsymbol{b}_4 &= [01100101]^T\end{aligned} \tag{6}$$

The next step is to select the most suitable system configuration. For this purpose, we use the qualitative evaluation procedure to determine the risk of the possible consequential states and to select the safest variant. In our application case we concentrate on the creation of the model on the basis of simplified system dependencies, because these can be

derived from the system representation, see figure 3. This approach of using the knowledge of the system structure as a basis is always possible independent of the data basis and the existence of any simulation. With this knowledge we can formulate simplified qualitative equations for each state of equation 3 in the form of:

$$\begin{aligned} T_c &= +T_{Act} - T_{ISFA} - T_{C1||C2} \\ \phi_c &= +\phi_{Act} - \phi_{ISFA} - \phi_{C1|C2} \\ \dot{V}_{AFS} &= +\dot{V}_{ISFA} - \dot{V}_{IRFA} \\ p_c &= +p_{ISFA} - p_{IRFA} \end{aligned} \tag{7}$$

The equations 7 still need to be converted to the QSIM notation to be used for qualitative evaluation. Therefore we define the qualitative behaviour of the system based on its constraints and introduce auxiliary variables. the qualitative constraints are then shown using the example of the cabin temperature T_c in equation 8:

$$\begin{aligned} &F^-(T_{ISFA}, b_{\text{ISFA}}), \\ &F^-(T_{CHX}, b_{\text{C1||C2}}), \\ &F^+(T_{Act}, Activity), \\ &ADD(T_{ISFA}, T_{CHX}, T_{nSum}), \\ &ADD(T_{Act}, T_{nSum}, T_c) \end{aligned} \tag{8}$$

In this case, the temperature T_c can be understood as the sum of the negative and positive effective parameters. On the one hand, the astronauts' activity lead to an increase in temperature, and on the other hand, the colder supply air through the ISFA and the cooling core work against it.

Finally if all qualitative system equations are defined, the list of reconfiguration options can be tested and validated. Based on current system data the qualitative variables can be initialised and the algorithm 1 can be executed. The QuAT which was introduced before can't be utilised for the visualisation of the system, because it is far too complex to present the results here in this place as it contains thousands of states. Instead the Table 2 shows a validation of the different reconfiguration options. For each of the state variables, which were defined in equation 3, we can create their own QuAT and analyse the predicted risk for each reconfiguration option. Overall this allows an estimation of how a specific configuration affects the different state variables and therefore an initial guess on which reconfiguration to apply. The total risk assumptions can be compared to suggest the option with the least transitions into risky operations.

An experienced operator might favour a configuration with a better performance for one specific state, based on the current fault diagnosis, but we select the option with minimal expected overall risk. In this case reconfiguration option $\boldsymbol{b}_3$ is considered optimal with the least totaled calculated risk. It performs well because the states $\dot{V}_{AFS}$ and p_c are not directly affected by the configuration changes and therefore exist in a steady state without further disturbance, and therefore without any expected risk. It is arguable whether the qualitative equation of $\dot{V}_{AFS}$ defined in equation 7 should be

	T_c	ϕ_c	$\dot{V}_{AFS}$	p_c
$\boldsymbol{b}_0$	42	42	33	38
$\boldsymbol{b}_1$	42	42	44	44
$\boldsymbol{b}_2$	34	34	38	33
$\boldsymbol{b}_3$	34	34	5	5
$\boldsymbol{b}_4$	42	42	38	33

Table 2: Risk score calculation for each reconfiguration option $\boldsymbol{b}_i$ based on the QuAT.

affected by the fanspeed of the CFA_1 or CFA_2, but as long as the cabin door is shut, the circulating air is only defined by the supply (ISFA) and return (IRFA) fan assemblies.

Measuring the effectiveness of qualitative state predictions is still an ongoing task in the project, but in its current form the supervision tool grants important insights by ranking the available reconfiguration options. For a given accident or failure multiple reconfiguration options can be identified, but in order to explicitly propose a solution and pave the way for autonomous deployment, a decision process must be integrated. By assessing the risk of upcoming qualitative states the decision can be forced to priories the well-being of the astronauts and a secure operation of the life support system.

5 Conclusion and future work

We present a novel approach that combines the fundamentals of qualitative system description with applications in artificial intelligence and system control theory. Our concept of qualitative prediction allows for the construction of an abstracted model based on fundamental knowledge of cause-effect relationships, enabling the prediction of complex system behaviour. Risk estimation plays a crucial role in selecting the appropriate configuration to recover from unintended system behaviour. However, the algorithm's performance currently hinders its application to systems with low response time. The combinatorial explosion of possible successor states is a computationally intensive task even with the proposed depth limitations. The operation time depends on factors such as the number of evaluated configurations, required depth, and the level of model detail. In our case the simplified qualitative equations in 7 analysed 2066 states in less than $30s$, increasing the amount of reconfiguration options to 10 increased the evaluation time to about $100s$ for roughly 7700 states and adding an additional state variables like the pressure at the first intersection increased the evaluation time to about $240s$. Of course the evaluation time depends on the used hardware, but the tendency is clear: Optimisation is necessary to improve the algorithm's efficiency.

The generation and definition of qualitative equations still requires expertise, and a poorly constructed model can limit overall functionality. Additionally, the abstract nature of qualitative solutions can pose challenges when converting them back into numerical contexts. To address these issues, the work of Say (Say and Kuru 1996) and Niggemann (Niggemann et al. 2021) shows promise in including system identification and merging of learned system behaviours, respectively. Incorporating these advancements

into our approach of constructing the qualitative analysis tool (QuAT) can enhance its capabilities. To build upon these ideas it might be worthwhile to include data based concepts to set probabilities for the state transitions to account for normal behaviour and the most probable transitions. This could help to predict the risk of an action more accurate, or rather to help to identify planned and safe transitions. On the other hand the probability for failures and anomalies can't be based on data-sets, if those issues only occur in rare instances especially if the supervision tool is meant to supervised data based methods.

Furthermore, the presented qualitative evaluation can be used in other tasks. The evaluation of predicted system states is of particular interest in the task domain of approaches based on neural networks. In this context, we want to research the possibility to apply the qualitative reasoning to reinforcement learning by integrating the prediction of expected system states as action masking in internal reward policies. With this approach risky actions will be avoided during training. By doing so, we hope to optimise the learning behaviour and drastically reduce learning effort.

6 Acknowledgments

This research is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU. We're grateful for the support.

References

Baheti, R. 2011. Cyber-physical systems. *The impact of control technology*, 12.1: 161–166.

Balzereit, K.; and Niggemann, O. 2022. AutoConf: New Algorithm for Reconfiguration of Cyber-Physical Production Systems. *IEEE Transactions on Industrial Informatics*, 19(1): 739–749.

Bredeweg, B. 2003. Current Topics in Qualitative Reasoning. *AI Magazine*, Volume 24(Number 4).

de Kleer, J. 1993. *A view on qualitative physics*. MIT Press Cambridge, MA.

de Kleer, J.; and Brown, J. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24: 7–83.

de Kleer, J.; and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence*, 32(1): 97–130.

Doyé, J. 2012. An Advanced Columbus Thermal and Environmental Control System. In *SpaceOps 2012 Conference*. Stockholm, Sweden: American Institute of Aeronautics and Astronautics.

Dvorak, D.; and Kuipers, B. 1989. Model-Based Monitoring of Dynamic Systems. In *IJCAI*, 1238–1243.

Garg, A.; Zhang, W.; Samaran, J.; Savitha, R.; and Foo, C.-S. 2022. An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6): 2508–2517. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Gheraibia, Y.; Kabir, S.; Aslansefat, K.; Sorokos, I.; and Papadopoulos, Y. 2019. Safety + AI: A Novel Approach to Update Safety Models Using Artificial Intelligence. *IEEE Access*, 7: 135855–135869. Conference Name: IEEE Access.

Kelm, B.; Balzereit, K.; Moddemann, L.; Myschik, S.; and Niggemann, O. 2022. Application of a Model-based Reconfiguration Approach for the ISS COLUMBUS Environmental Control and Life Support System (ECLSS). *Proceedings of the 33nd International Workshop on Principle of Diagnosis, Toulouse, France*.

Klenk, M.; De Kleer, J.; Bobrow, D.; and Janssen, B. 2014. Qualitative reasoning with modelica models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Kuipers, B. 1986. Qualitative Simulation. *Artificial Intelligence*, 29(3): 289–338.

Niggemann, O.; Stein, B.; Vodencarevic, A.; Maier, A.; and Kleine Büning, H. 2021. Learning Behavior Models for Hybrid Timed Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1): 1083–1090.

Salles, P.; and Bredeweg, B. 2006. Modelling population and community dynamics with qualitative reasoning. *ecological modelling*, 195(1-2): 114–128.

Say, A.; and Kuru, S. 1996. Qualitative system identification: deriving structure from behavior. *Artificial Intelligence*, 83(1): 75–141.

Simon, H. A. 1991. *Qualitative simulation modeling and analysis*, volume 5. Springer-Verlag.

Struss, P.; and Price, C. 2003. Model-based systems in the automotive industry. *AI magazine*, 24(4): 17–17.

Tjoa, E.; and Guan, C. 2020. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11): 4793–4813.

Tjoa, E.; and Guan, C. 2021. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11): 4793–4813. ArXiv:1907.07374 [cs].

Trave-Massuyes, L.; Ironi, L.; and Dague, P. 2003. Mathematical foundations of qualitative reasoning. *AI magazine*, 24(4): 91–91.

Wan, A.; Dunlap, L.; Ho, D.; Yin, J.; Lee, S.; Jin, H.; Petryk, S.; Bargal, S. A.; and Gonzalez, J. E. 2020. NBDT: neural-backed decision trees. *arXiv preprint arXiv:2004.00221*.

Wan, A.; Dunlap, L.; Ho, D.; Yin, J.; Lee, S.; Jin, H.; Petryk, S.; Bargal, S. A.; and Gonzalez, J. E. 2021. NBDT: Neural-Backed Decision Trees. ArXiv:2004.00221 [cs].

Williams, B. C.; Ingham, M. D.; Chung, S.; Elliott, P.; Hofbaur, M.; and Sullivan, G. T. 2003. Model-based programming of fault-aware systems. *AI Magazine*, 24(4): 61–61.

Wolf, W. 2009. Cyber-physical systems. *Computer*, 42(03): 88–89.