



A Causal Model-based Planner for the Reconfiguration of Continuous Processes

Reinartz, Christopher Clarc; Enevoldsen, Thomas Thuesen; Galeazzi, Roberto; Ravn, Ole

Published in:
Proceedings of European Control Conference 2021

Link to article, DOI:
[10.23919/ECC54610.2021.9655016](https://doi.org/10.23919/ECC54610.2021.9655016)

Publication date:
2021

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Reinartz, C. C., Enevoldsen, T. T., Galeazzi, R., & Ravn, O. (2021). A Causal Model-based Planner for the Reconfiguration of Continuous Processes. In *Proceedings of European Control Conference 2021 IEEE*.
<https://doi.org/10.23919/ECC54610.2021.9655016>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Causal Model-based Planner for the Reconfiguration of Continuous Processes

Christopher Reinartz¹, Thomas Thuesen Enevoldsen¹, Roberto Galeazzi¹ and Ole Ravn¹

Abstract—This paper presents a planning framework for discrete planning of human operations in highly automated industrial plants, where manual and automatic control coexist. Based solely on qualitative knowledge of the controlled continuous process represented through signed directed graphs, the planner exploits a greedy algorithm to determine the optimal action for the human operator upon changes in the process' operating conditions or reconfiguration of the control system due to faulty conditions. The planner results in a decision support tool that instructs the human operator on the best course of action. The planning framework and the resulting planner are demonstrated on a quadruple-tank system.

I. INTRODUCTION

The operation of large industrial plants involves the interaction between human operators and automated control systems. The goal is to ensure production on a high level of consistency, safety and economic efficiency that cannot be achieved by either manual or fully automatic control. Unit operations are commonly controlled using distributed control schemes, with Supervisory Control and Data Acquisition (SCADA) systems enabling human operators to issue process commands. In nominal operation, the operator configures the system for maximum profitability. In the event of abnormal system behaviour the operator ensures the return to safe operation or initiates the shut-down of the plant. Current trends towards increased automation and decreased human supervision of process plants challenge this established method of operation. Increased automation entails increased instrumentation and monitoring, which, in theory, improves plant supervision capabilities. In practice, the amount of monitoring exceeds what an operator can process, leading to cognitive overload that can result in slow or incorrect decision making, especially in unforeseen situations. Automated condition monitoring and decision support systems can be used to reduce the occurrence of such situations by providing contextual information that facilitates the decision making. The last decades have seen significant progress in both quantitative model-based and data-driven fault diagnosis, but the scalability of quantitative process models to plant-scale and the lack of interpretability of results generated from data-driven methods remain known drawbacks. The demand for methods that yield explainable results on a plant-scale requires re-evaluation and extension of approaches that were previously shown to be capable of

satisfying these properties, such as causality-based diagnosis and reasoning systems [1].

Causal process models have been considered for root-cause analysis and process supervision for a long time [2], and nowadays, improved automated generation and validation techniques enable their application for a wide range of processes. [3] presents the generation of causal models in form of hierarchically structured directed graphs based on transfer entropy, while [4] uses a Bayesian model to identify optimal causal network structures for root-cause diagnosis. A method for semi-automated generation of causal functional models from existing documentation of plant topology in the form of P&ID diagrams is presented in [5]. [6] presents how cross-correlation and transfer entropy can be used to validate manually or semi-automated models using process data.

The development of applications utilizing causal models proceeded concurrently to the development of new modeling strategies with a strong focus on techniques for root-cause analysis of process faults. [4], [7] and [8] use causal graphs for fault detection and root-cause analysis of a chemical engineering benchmark process, highlighting the capability of causal models to diagnose faults for which no historic process data is available. The recent successes in the generation of causal models and their use in process diagnosis promotes an investigation of their further applicability for decision support, especially in areas where other approaches perform poorly, such as the automatic actuation and reconfiguration of complex processes with limited quantitative knowledge. A scheduling approach for coke gas systems based on a joint application of causal-models and least-squares support vector machines is proposed by [9].

A. Novelty and contribution

The paper proposes a novel planning framework for the operation of continuous processes during transitions between steady-states, where automatic and manual control contribute to the regulation. A causal model in form of a signed directed graph is created to describe the qualitative knowledge about the impact of manual and automatic control actions onto the process variables. This is encoded in an effect matrix generated through an algorithm that propagates steady-state process responses to disturbances and deliberate state changes in the presence of control loops. A greedy planner is adopted to find the locally optimal operator action based on the minimization of a cost functional. The planner thus results in a decision support tool for operators that suggests the best course of action during changes in operating points or process reconfiguration due to the occurrence of faults.

*This work is supported by the Danish Hydrocarbon Research and Technology Centre.

¹Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark {ccrein, tthen, rg, or}@elektro.dtu.dk

II. PLANNING FRAMEWORK

To design a planning framework that can mimic an operators role in process supervision, the planning environment has to be set-up, such that the operators perception of the system and his/her means of influence are represented. It is assumed that the system under investigation is a continuous process with input-dependent stationary operating points and that the operator has complete knowledge of the means of influencing the process, e.g. set-point changes or manual adjustment of manipulated variables, and that he/she has at least qualitative knowledge about the effects of said means.

The planning framework should be capable of defining a plan based on qualitative effects, while remaining open to extensions that integrate quantitative knowledge, such as transition times and magnitudes. To remain open to such extensions, the planning environment has to be quantitative rather than qualitative. The following sections define the main constituents of the planning framework.

A. State space

The state of a process, as an operator can perceive it, is defined by the available process variables, meaning measurements and manipulated variables. Each process variable is bounded with known bounds. The state x is defined as the vector containing a specific value for each process variable $x_i \in [x_{i,min}, x_{i,max}]$. The state space $X \subseteq X_B = \{[x_1, \dots, x_n] | x_i \in [x_{i,min}, x_{i,max}]\}$ is the set of all possible states the system can assume. If unity-based normalization is applied to the process variables, the state space can be defined as $X \subseteq X_B = [0, 1]^n$, where n is the number of process variables. The state space X is the subset of all valid process configurations within the n -dimensional cubic space X_B that is formed by the bounds of the process variables or, if only steady state operation is considered, the subset of all valid steady state configurations of the process. It is not possible to determine the exact shape of the subset without quantitative knowledge of the system dynamics. For steady state analysis it is possible to infer that the state space forms a p -dimensional manifold within X_B , where p is equal to the degrees of freedom of the process. This follows directly from the observation that for each unique configuration of independent manipulated variables, and thus degrees of freedom, the process will assume a unique stationary operating point, if the manipulated variables themselves are included in the definition of the state.

B. Action space

The action space U contains all actions available to the operator in a given situation. Some actions might only be available for specific process configurations, resulting in a state-dependent action space. Every action is defined as a tuple (u'_i, δ_{u_i}) of action identifier and action magnitude. A typical example of an operator action, taking manual control of a valve position, could e.g. be described by the tuple ('open valve', 10%). The operator has the option of either increasing or decreasing any manipulated variable, including control set-points, in the process. While decreasing

a manipulated variable can be represented by a negative δ_{u_i} , it was decided to create separate actions for increasing and decreasing actions, thus only allowing $\delta_{u_i} > 0$. The magnitude of each action is bounded by the difference between the current state of the manipulated variable and its maximum or the current state and its minimum for increasing and decreasing actions, respectively.

C. Representing qualitative effects

In the context of this paper, a qualitative effect is defined as an effect with a known direction of change but unknown magnitude, which can be described by non-deterministic state transitions [10]. Considering future extensions of the planner that feature quantitative information, describing state transitions non-deterministically seems impractical. Instead, a deterministic state transition model that uses process feedback to account for uncertainty about the effect-magnitude is used in this contribution.

D. State-transition function

The qualitative effects of all actions are summarized in an effect matrix $G(x) = [(g^T(x, u_1), \dots, g^T(x, u_m))]^T$, an $m \times n$ matrix for action and state spaces of m and n dimensions, respectively. The state transition, without explicitly modeling uncertainty, is written as $x' = f(x, u) = x + \delta_{u_i} g^T(x, u)$. The uncertainty about the state transition can be modeled explicitly as a nature action θ , which is a natural extension of the presented planning framework, but not considered further in this contribution.

E. Effect matrix

As mentioned in Section II-D, the effect matrix summarizes the qualitative effect of the actions in the action space on all variables in the state vector. Considering only qualitative effects, actions must either increase or decrease variables or leave them unaffected, so that the configuration s of any process variable following an action must be within the set $\{1, 0, -1\}$, where 1, 0 and -1 encode "increased", "not affected" and "decreased", respectively. Given a causal model of a system in form of a signed directed graph $G = (V, E)$, for which the set of vertices V includes the measured and manipulated process variables and the set of signed edges E describes their causal relations, the propagation of effects originating in single process variables through the system can be estimated using graph propagation, as shown by [11]. Simple graph propagation, however, does not consider the influence of control systems correctly and thus leads to spurious results. Different approaches to describe control systems in causal models have been presented in e.g. [3] and [12], but typically involve additional modeling effort. The propagation strategy presented in this paper in Algorithm 1 correctly derives the steady-state development of variables in closed-loop systems given knowledge of the active control-loops. Control loops are indicated in the causal model by marking controlled variables as "controlled" and by labeling the edges from controlled to manipulated variables as "control edges". It is further assumed that the

Algorithm 1: propagate_configuration

Given: $G = (V, E)$, $v_{start} \in V$, $s(v_{start}) \in \{-1, 0, 1, \{\}\}$

```

1  $A \leftarrow \{v_{start}\}$ ,  $D \leftarrow \{v_{start}\}$ 
2 while  $A \neq \{\}$  do
3    $v \leftarrow A.pop\_first()$ 
4   if  $is\_controlled(v)$  then
5      $e \leftarrow find\_control\_edge(G, v)$ 
6      $(v, w) \leftarrow e$ 
7     if  $w \notin D$  then
8        $s(w) \leftarrow s(v) \cdot sgn(e)$ 
9        $s(v) \leftarrow 0$ 
10       $D \leftarrow D \cup w$ 
11       $A.append(w)$ 
12      continue
13   forall  $\{e = (u, w) \in E | u = v\}$  do
14     if  $w \notin D$  then
15        $s(w) \leftarrow s(u) \cdot sgn(e)$ 
16        $D \leftarrow D \cup w$ 
17        $A.append(w)$ 
18 forall  $v \in V$  do
19   if  $s(v) = \{\}$  then
20      $s(v) \leftarrow 0$ 

```

configuration of a node is determined through the shortest path from the root of the propagation, which is enforced by maintaining a set D of already visited “dead nodes” while propagating using a breadth first approach that maintains a list of active nodes A . An example of the derivation of an effect matrix is provided in Section IV-B.

F. Representing time

The inclusion of qualitative effects in the action space entails that the duration of the actions can typically not be estimated. Thus, it is sufficient to represent the sequence of actions by an increasing stage index k .

G. Formalization

Having introduced the different components of the planner, we can now formalize the planning framework. The *state space* X is a nonempty, uncountably infinite set representing all possible states the system under investigation could assume. The *action space* U is a finite, nonempty set of, possibly state dependent, actions. The action space includes a termination action u_T , which does not result in a state change and is applied when the current state is within the goal set X_G . The *state transition function* yields a new state, given the current state and action. A *set of stages* k denotes the relative temporal ordering of the states that are part of the devised plan. An explicit *initial state* $x_I \in X$ is defined for each application of the planning problem. Different from the initial state, the *goal set* $X_G \subset X$ is defined as a continuous region. The *cost* L as a function of a state history \tilde{x}_F and action history \tilde{u}_K is defined as the sum of the state and action dependent transition costs $l(x, u)$ and the cost associated with the state that is reached after the execution of the plan, which is defined as a potential function $\phi(x)$ on the state space.

- 1) *State space:* $X \subseteq X_B = [x_{i,min}, x_{i,max}]^n$
- 2) *Action space:* $U(x) = [(u_1, \delta_{u,1}), \dots, (u_m, \delta_{u,m}), u_T]$

3) State transition function:

$$x' = f(x, u) = x + \delta_u \cdot g^T(x, u)$$

4) Stages: $k = 1, 2, \dots, K + 1, F = K + 1$

5) Initial state: $x_I \in X$

6) Goal set: $X_G \subset X$

7) Cost functional:

$$L(\tilde{x}_F, \tilde{u}_K) = \sum_{k=1}^K l(x_k, u_k) + \phi(x_F)$$

$$l(x_k, u_k) \geq 0$$

$$\phi(x) \geq 0, \forall x \in X \quad \phi(x) = 0, \text{ iff } x \in X_G$$

III. QUADRUPLE-TANK SYSTEM

The quadruple-tank system is a multivariate laboratory process presented by [13]. The process consists of four water tanks, which are setup as shown in Fig. 1, where the water level h_i for all tanks is defined on the interval $[0, 0.2]$ meters, the valve voltages v_1 and v_2 between $[0, 6]$ volts, and the relative valve positions γ_1 and γ_2 between $[0, 1]$.

The system is described by the following set of equations

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1} \sqrt{2gh_1} + \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1 \quad (1)$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2 \quad (2)$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3} \sqrt{2gh_3} + \frac{(1 - \gamma_2) k_2}{A_3} v_2 \quad (3)$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4} \sqrt{2gh_4} + \frac{(1 - \gamma_1) k_1}{A_4} v_1, \quad (4)$$

The simulations are based on the controller and plant parameters specified in [13], with additional outputs for the water levels h_3 and h_4 . A clamping-based anti-windup scheme is added to improve the performance of the integrator of the implemented PI-controllers after extended periods of non-rejectable steady-state error.

The voltages v_1 and v_2 are used by PI regulators to regulate the water levels h_1 and h_2 , while the valve positions γ_1 and γ_2 are manipulated variables available to the human operator to act on the process. When modifying the valve positions, it has to be considered that the system is only minimum phase for $\gamma_1 + \gamma_2 > 1$, and that control performance degrades for other configurations. Measurement noise on the four tank levels is introduced as white Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.25\text{mm}$.

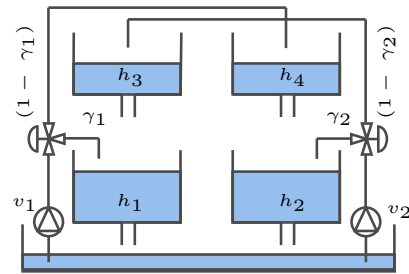


Fig. 1: Quadruple-tank system

IV. CASE-STUDY

The presented planning framework combined with a greedy planner, which operates using a strict best-first approach [10], is applied to the quadruple-tank system. The

modeling uncertainty is expressed implicitly by using immediate state-feedback after a one-stage planning process. To emulate a human operator in the loop, a reference signal generator is used to translate the actions recommended by the greedy planner into set-points following a ramp profile for the control-references and valve positions. The slopes for the reference signals are set as $2.5e^{-4}ms^{-1}$ and $2e^{-4}s^{-1}$ for the water levels and valve positions, respectively to consider the dynamics of the controlled system.

A. State and action space

The eight process variables $h_1, h_2, h_3, h_4, v_1, v_2, \gamma_1, \gamma_2$ define a state space $X = [0, 0.2]^4 \times [0, 6]^2 \times [0, 1]^2$, based on the bounds of the water levels, voltages and valve positions. Applying normalization, the state space becomes $X_{norm} \subset [0, 1]^8$. In case of nominal closed-loop control, an operator can influence the process by changing set-points for h_1 and h_2 and manually interfacing with the process through γ_1 and γ_2 , so that the set of actions becomes $U = \{h_1^+, h_2^+, \gamma_1^+, \gamma_2^+, h_1^-, h_2^-, \gamma_1^-, \gamma_2^-\}$, where the superscripts '+' and '-' denote "increase" and "decrease". The magnitude of all actions in the normalized space is set to $\delta_u = 0.005$, so that the state transition function described in Section II-D becomes $f(x, u) = x + 0.005g^T(x, u)$.

B. Causal graph and effect matrix

The causal model for the quadruple-tank system, represented by the signed directed graph shown in Fig. 2, is generated from (1)-(4) using the procedure shown by [11] for the extraction of signed directed graphs from differential equations. For the differential equation

$$\frac{dx_i}{dt} = f_i(x_1, \dots, x_n), \quad (5)$$

the sign of edges of causally connected variables is given by

$$\text{sgn}(x_j \rightarrow x_i) = \text{sgn} \left[\frac{\partial f_i}{\partial x_j} \Big|_{x_1^0, \dots, x_n^0} \right], \quad (6)$$

where "sgn($x_j \rightarrow x_i$)" is the sign of the edge from x_j to x_i . Graphs derived in this manner typically include both cycles and self-cycles if applied to the analysis of continuous processes. Cycles occur when closed-loop control systems or recycle streams are represented, while self-cycles arise if

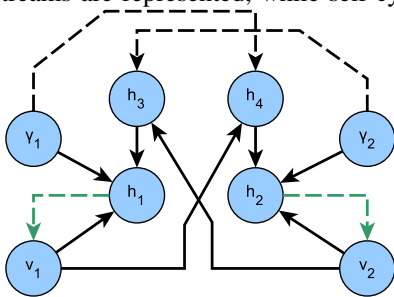


Fig. 2: Signed directed graph representing the qualitative causal relations of the quadruple-tank system in closed-loop. Green edges represent information transfer inside control loops. Self-loops are not shown. Positive and negative edges are indicated by continuous and dashed lines, respectively.

the derivative of a variable directly depends on the variable itself, as e.g. the water levels h_1-h_4 in (1)-(4). We rely on Algorithm 1 to correctly describe the effect of loops and self-loops for steady-state propagation, resulting in the following effect matrix for the system in closed-loop operation

$$G_{cl}(x) = \begin{matrix} h_1^+ \\ h_2^+ \\ \gamma_1^+ \\ \gamma_2^+ \\ h_1^- \\ h_2^- \\ \gamma_1^- \\ \gamma_2^- \end{matrix} \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & v_1 & v_2 & \gamma_1 & \gamma_2 \\ 1 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & -1 \end{pmatrix}. \quad (7)$$

To clarify, the steady-state propagation of an increase of the control set-point for the tank level h_1 , summarized in the first row of (7), is described. The graph propagation is initialized by setting the configuration of the manipulated variable belonging to the control-loop of tank 1 to $s(v_1) = 1$ and marking node v_1 as a dead node. The graph propagation along the edges (v_1, h_1) and (v_1, h_4) yields $s(h_1) = 1$ and $s(h_4) = 1$. In the subsequent iteration, the control edge (h_1, v_1) is not propagated because v_1 was previously marked as a dead node, but the propagation along (h_4, h_2) results in $s(h_2) = 1$. Because h_2 is a controlled variable, the corresponding subroutine in Algorithm 1 is activated, resulting in the assignment $s(v_2) = -1$ through the control edge (h_2, v_2) and subsequent reassignment of $s(h_2) = 0$, indicating that the configuration of h_2 will not change once dynamics have settled, because the controller eventually rejects the disturbance. The final assignment yields $s(h_3) = -1$ along the edge (v_2, h_3) , since $s(v_2) = -1$ was set in the previous iteration. The iteration terminates following the assignment of h_3 , because there are no edges leading to nodes that are not marked as dead from h_3 , leading to the default assignments $s(\gamma_1) = 0$, $s(\gamma_2) = 0$ for the remaining two variables.

C. Cost function

It is assumed that state transitions entail no cost, so that the cost of an action is defined as the cost of the resulting state, $\phi(x')$. The potential function serving as a heuristic for the cost of a state, $\phi(x)$, is defined as the Euclidean distance between the state and an ideal goal state $x_G \in X_G$, i.e. $\phi(x) = \|x - x_G\|_2$. The cost function is not applied to the full state vector, but to a subset of goal variables of the state vector for which the goal state is known prior to the execution of the plan. For the quadruple-tank system, a typical subset of goal variables is $\{h_1, h_2, h_3, h_4\}$. The termination criterion is met if the state lies within the defined goal set.

D. Greedy algorithm

The implemented planner searches using a greedy best-first approach, as presented by [10]. At each planning step, m potential new states are evaluated, where m is the number of applicable actions in the current measured state x . The locally optimal action $u_{best} = \arg \min_u (l(x, u) + \phi(g(x, u)))$ is chosen for execution until the next iteration of the planning algorithm.

E. Scenarios

Scenario 1, displayed in Fig. 4 describes the planning for a change in operating point during nominal operation that cannot be handled by changing only the control inputs v_1 and v_2 for h_1 and h_2 , but requires process modifications outside the nominal control scheme. Scenario 2 describes the same change in operating point as Scenario 1, and additionally imposes constraints on the valid region of the state space, requiring that the water level $h_2 < 0.11m$, until the level in tank 1 satisfies $h_1 > 0.13m$. This constraint is imposed by defining a rectangular obstacle region $X_{obstacle} = [0m, 0.13m] \times [0.11m, 0.2m]$ in the h_1, h_2 subspace of the state space, which has to be avoided by the planner. This scenario is meant to demonstrate how operational constraints can be represented in the planning framework. Fig. 3 shows the different developments of the water levels in tanks 1 and 2 that result from the additional constraints. Scenario 3 describes a reconfiguration of the process in case of a diagnosed partial failure of pump 1. The pump capacity is reduced by 75% at $t = 200s$, leading to a situation in which the desired set-points of the water levels in the bottom tanks cannot be reached and in which h_1 is no longer controlled, since pump 1 constantly operates at maximum efficiency. This new configuration is accounted for by removal of the control edge (h_1, v_1) in the causal graph and calculation of a new effect matrix using Algorithm 1, which yields

$$G_{pf}(x) = \begin{matrix} h_2^+ \\ \gamma_1^+ \\ \gamma_2^+ \\ h_2^- \\ \gamma_1^- \\ \gamma_2^- \end{matrix} \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & v_1 & v_2 & \gamma_1 & \gamma_2 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 1 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 & -1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}. \quad (8)$$

The goal of the planning algorithm is to reconfigure the system such that the water levels in the bottom tanks are returned to the states they were at before the pump failure. Fig. 5 shows the reconfiguration of the system after the activation of the planner at $t = 1000$ to the desired goal state by compensating for the lack of pump 1 with an increase of the flow through pump 2 and an adjustment of the three-way valves γ_1 and γ_2 . The initial and ideal goal states for the scenarios are summarized in Table I. The set of relevant goal variables is reduced from $\{h_1, h_2, h_3, h_4\}$ in the first two scenarios to $\{h_1, h_2\}$ in the third scenario to account for the reduced number of degrees of freedom resulting from the partial pump failure. The bounds of the goal sets X_G are shown as dashed grey lines in Fig. 3, 4 and 5.

TABLE I: Initial and goal states for the tested scenarios. ‘-’ indicates variables not included in the goal set.

	h_1	h_2	h_3	h_4	v_1	v_2	γ_1	γ_2
$x_{I,S1,2}$	0.1	0.1	0.0025	0.05	4.20	1.16	0.59	0.59
$x_{G,S1,2}$	0.15	0.15	0.001	0.0025	-	-	-	-
$x_{I,S3}$	0.16	0.16	0.004	0.07	5.26	1.53	0.6	0.6
$x_{G,S3}$	0.16	0.16	-	-	-	-	-	-

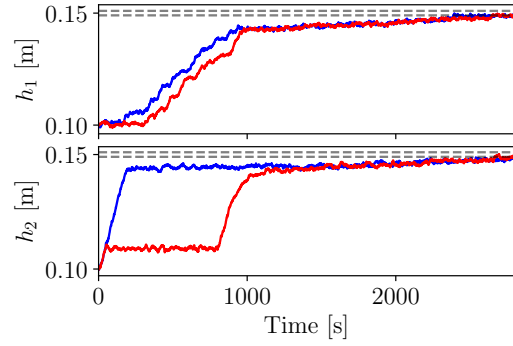


Fig. 3: Comparison between Scenario 1 (blue) and 2 (red), representing a change of operating point without and with constraints. Dashed lines indicate the goal region.

F. Results

The planner finds a feasible path from the initial to the goal configuration for all presented scenarios. Scenario 2 illustrates that operational constraints can be considered as part of the planning scheme as long as they can be formulated as obstacles in the state space. Scenario 3 shows that the planning framework is capable of re-configuring the system in a fault scenario, if it is provided with the information about the source of the fault. In practice, this requires a fault diagnosis scheme which is capable of fault isolation.

G. Discussion

The planner activates in three situations: (i) the operator changes the desired process operating-point, i.e. a new goal set X_G is defined; (ii) a fault is diagnosed which determines the need of combined manual and automated action to steer the process into the goal set; (iii) a disturbance enters the control loop and the automatic control action does not suffice to keep the process in the goal region. At the stage k the planner assesses if the state is in the goal region and suggests the locally optimal action $u_k \in U$ to the human operator. The operator should then execute the suggested action and leverage his/her expert knowledge to quantify its magnitude. At the next iteration, based on the received feedback from the process, the planner re-assesses the situation and suggests a new locally optimal action $u_{k+1} \in U$, potentially confirming the previous one, i.e. $u_{k+1} = u_k$, to the operator. The frequency upon which the planner is run is a tuning parameter that depends on the dominant time constants of the continuous process to be regulated.

While the results are promising, using a greedy planner does not guarantee that a feasible path is found since there is the risk of “getting stuck” in a local minimum of the potential function $\phi(x)$. This problem could be addressed by using tabu search or by combining greedy search with random walks. Further, the planner may output a less reliable plan for systems with fast transient dynamics since the underlying state transition model is derived for steady-state configurations.

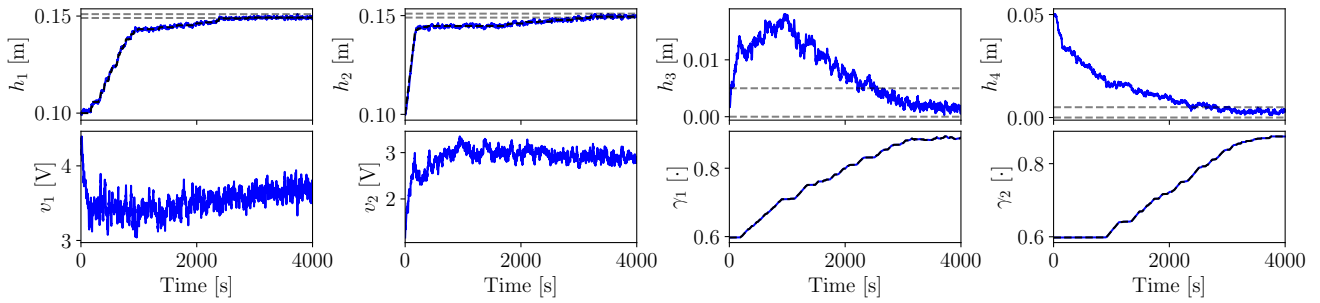


Fig. 4: Scenario 1: The control set-points and valve positions are changed to achieve a transition between the initial and final steady-state operating conditions described in Table I. Dashed lines indicate the suggested reference and γ changes. Blue lines indicate the system trajectories. Dashed grey lines indicate the goal region of the planner.

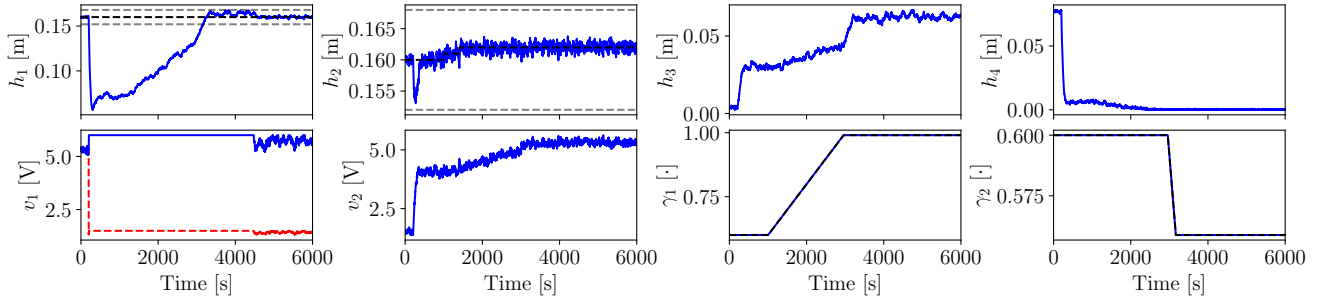


Fig. 5: Scenario 3: Dashed lines indicate the suggested reference and γ changes. Blue lines indicate the system trajectories. The dashed red line indicate the true performance of v_1 once the fault occurs at $t = 200s$. The reconfiguration of the system is initiated at $t = 1000s$. Transparent dashed lines indicate the goal region of the planner.

V. CONCLUSIONS

A framework for discrete planning of human operations in automated continuous processes is presented. The framework uses causal models in the form of signed directed graphs to represent the effect of manual and automatic control actions onto the process variables at steady state. This qualitative knowledge is encoded in an effect matrix that underpins the state transition function utilized to explore the state space when the process changes its point of operation or needs to reconfigure due to faults. The planning framework can accommodate variable levels of quantitative knowledge about the continuous process and can generate plans of variable length. The planner's performance using models that assume very limited system knowledge was successfully demonstrated on the quadruple-tank system, utilizing a greedy search algorithm with a single-stage planning horizon (one operator action) to compensate for modeling uncertainty. The proposed causal model-based planner works as a decision support tool for control room operators by providing a qualitative proposal about the best course of action to regain desired steady state operation after intentional or unexpected changes of the process operating point.

REFERENCES

- [1] V. Venkatasubramanian, "The promise of artificial intelligence in chemical engineering: Is it here, finally?" *AIChE Journal*, vol. 65, no. 2, pp. 466–478, 2019.
- [2] L. Leyval, S. Gentil, and S. Feray-Beamont, "Model-based causal reasoning for process supervision," *Automatica*, vol. 30, no. 8, pp. 1295–1306, 1994.
- [3] R. Suresh, A. Sivaram, and V. Venkatasubramanian, "A hierarchical approach for causal modeling of process systems," *Computers and Chemical Engineering*, vol. 123, pp. 170–183, apr 2019.
- [4] J. Mori, V. Mahalec, and J. Yu, "Identification of probabilistic graphical network model for root-cause diagnosis in industrial processes," *Computers and Chemical Engineering*, vol. 71, pp. 171–209, 2014.
- [5] M. Song and M. Lind, "Towards Automated Generation of Function Models from P&IDs," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 1081–1084.
- [6] F. Yang, S. Shah, and D. Xiao, "Signed directed graph based modeling and its validation from process knowledge and process data," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 41–53, 2012.
- [7] Y. Wan, F. Yang, N. Lv, H. Xu, H. Ye, W. Li, P. Xu, L. Song, and A. K. Usadi, "Statistical root cause analysis of novel faults based on digraph models," *Chemical Engineering Research and Design*, vol. 91, no. 1, pp. 87–99, 2013.
- [8] B. He, T. Chen, and X. Yang, "Root cause analysis in multivariate statistical process monitoring: Integrating reconstruction-based multivariate contribution analysis with fuzzy-signed directed graphs," *Computers & chemical Engineering*, vol. 64, pp. 167–177, may 2014.
- [9] F. Jin, Z. Lv, M. Li, L. Mou, J. Zhao, and W. Wang, "A causal model-based scheduling approach for coke oven gas system in steel industry," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 7 – 12, 2018, 5th IFAC Workshop on Mining, Mineral and Metal Processing MMM 2018.
- [10] S. M. LaValle, "Planning algorithms," *Planning Algorithms*, vol. 9780521862, pp. 1–826, 2006.
- [11] F. Yang, S. L. Shah, and D. Xiao, "Signed directed graph-based hierarchical modelling and fault propagation analysis for large-scale systems," *IET Control Theory & Applications*, vol. 7, no. 4, pp. 537–550, 2013.
- [12] M. R. Maurya, R. Rengaswamy, and V. Venkatasubramanian, "A signed directed graph-based systematic framework for steady-state malfunction diagnosis inside control loops," *Chemical Engineering Science*, vol. 61, no. 6, pp. 1790–1810, 2006.
- [13] K. H. Johansson and J. L. R. Nunes, "A multivariable laboratory process with an adjustable zero," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, 2000.