Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 4, 561–568
DOI: 10.2478/v10006-008-0049-0

SUBOPTIMAL FAULT TOLERANT CONTROL DESIGN WITH THE USE OF DISCRETE OPTIMIZATION

ZDZISŁAW KOWALCZUK, KRZYSZTOF E. OLIŃSKI

Department of Decision Systems
Gdańsk University of Technology, ul. Narutowicza 11/12, 80–952 Gdańsk, Poland
e-mail: {kova,kolin}@pg.gda.pl

This paper presents a concept of designing fault tolerant control systems with the use of suboptimal methods. We assume that a given (nonlinear) dynamical process is described in a state space. The method consists in searching (at the off-line stage) for a trajectory of operational points of the system state space. The sought trajectory can be constrained by certain conditions, which can express faults or failures already detected. Within this approach, we are able to use the autonomous dynamics of the process in order to minimize a control cost index (a sub-optimality property). The search itself is based on finding a cheapest path in a graph structure, which represents the system's dynamics described in the state space. Such a cheapest path (if it exists) represents the sought trajectory. Another (on-line) design stage consists in tracking this trajectory by an executive controller.

Keywords: optimal control, fault tolerant systems, nonlinear models, operations research, discrete optimization.

1. Introduction

In this paper, without making a distinction between a *fault* and a *failure* (Chowdhury and Chen, 2006), we shall differentiate between faults that are *admissible* and *inadmissible* during the system operation. Dealing with admissible faults implies modifying an original process model so as to take into consideration all effects invoked by these faults. The occurrence of any fault from the group of inadmissible faults requires introducing some additional constraints for the sought trajectory of the operational points of the process. The subject of this work can thus be placed within the field of fault tolerant control, FTC (Zhang, 2007). We assume that a given dynamical process is represented by a state-space model, which describes both autonomous and forced dynamics of the process. Such a model can be either linear or nonlinear with soft and hard nonlinearities (equivalent to the lack of derivatives of a function in some points of its domain).

During the first, off-line stage, we try to find a trajectory in the state space of the process model, which should satisfy two principal conditions (Kowalczuk, Rudzinska-Kormanska and Olinski, 2007; Kowalczuk and Olinski, 2007). Firstly, the sought trajectory circumvents the areas (called forbidden zones) associated, for instance, with inadmissible faults previously detected and identified. Secondly, we can utilize the autonomous dynamics of the process in order to minimize an assumed control cost indicator.

An operational workspace, being a subset of the state space, is composed of a set of segments. For each segment, a set of representative values, which reflect the properties of the process dynamics within the area of a given segment, is determined.

The searching procedure applied is based on finding a cheapest path in a graph structure (Kowalczuk *et al.*, 2007; Kowalczuk and Olinski, 2007), which represents the dynamics of the process described in the state space.

A discrete optimization algorithm is utilized in searching the state-space graph for the cheapest path connecting the nodes containing the initial and terminal points of the sought trajectory.

The second, on-line executive stage consists in tracking the reference trajectory by an executive controller.

2. Formal description

We start the general problem formulation by defining the *autonomous* and *forced* components of the dynamics of a given process. Let us assume that the function which

describes the process dynamics has the following form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{1}$$

and belongs to the class

$$\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n. \tag{2}$$

Consider the following decomposition of (1) into two parts:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_x + \dot{\mathbf{x}}_u = \mathbf{f}_x(\mathbf{x}, t) + \mathbf{f}_u(\mathbf{x}, \mathbf{u}, t), \tag{3}$$

where $\mathbf{f}_u(\mathbf{x}, 0, t) = 0$. With the above, let us introduce the following nomenclature:

- $\dot{\mathbf{x}}_x = \mathbf{f}_x(\mathbf{x}, t)$ represents the *autonomous dynamics*,
- $\dot{\mathbf{x}}_u = \mathbf{f}_u(\mathbf{x}, \mathbf{u}, t)$ portrays the *forced dynamics*,
- $\mathbf{x}_x$,$\mathbf{x}_u$ are their two corresponding *state contributors*.

Let us now present several definitions (Kowalczuk *et al.*, 2007; Kowalczuk and Olinski, 2007) vital for this work.

Definition 1. Any sequence $E(\mathbf{u})$ of consecutive states of a given dynamical system for a feasible control input $\mathbf{u}$ from a set $\mathcal{U}$ is said to be a *system's trajectory* or a *trajectory of operational points* of the system state space.

Definition 2. A bounded subset P of the state space in the form of a hypercube in $\mathbb{R}^n$, which is taken into account while seeking an optimal trajectory, is said to be an *operational workspace*.

Definition 3. A subset Z of P prohibited for operational points is referred to as a *forbidden zone*. This means that the sought optimal trajectory cannot enter it.

Definition 4. A *transition vector* Λ is an ordered set of two elements $\{x_0, x_k\}$:

$$\Lambda = \{(x_0, x_k) : x_0, x_k \in E(\mathbf{u})\}, \tag{4}$$

where x_0 is the first element and x_k is the last element of the sought optimal trajectory.

Definition 5. Let $\mathcal{A}$ be a set of all trajectories $E \in \mathcal{A} \subset P$. Any real function of the class $\mathcal{A} \to \mathbb{R}$ is said to be a *cost function* $J(E(\mathbf{u}))$ of these trajectories.

Definition 6. Let $\Xi \subset \mathcal{A}$ be a subset of all possible trajectories which start at a given point x_0 and terminate at x_k. We say that a trajectory E^* is *optimal* if it satisfies the following conditions:

$$J(E) \geq J(E^*), \quad \forall E \in \Xi, \tag{5}$$

$$\mathbf{x} \in P \setminus Z, \quad \forall \mathbf{x} \in E^*. \tag{6}$$

Definition 7. *The segmentation* of a given operational workspace P into a set of N_s segments is defined as

$$P = \bigcup_{j=1}^{N_s} \Phi_j, \tag{7}$$

where each pair of segments also has to satisfy the following separability condition:

$$(\Phi_i - \delta\Phi_i) \cap (\Phi_j - \delta\Phi_j) = \emptyset \tag{8}$$

for all $i, j = 1, \ldots, N_s; i \neq j$, where δ denotes the closure of a given set.

Definition 8. Any quantity used in optimization and assigned to a segment is said to be a *representative*.

Definition 9. A flow graph portraying the arrangement of segments of a given operational workspace P is said to be a *state-space graph*. According to the respective definitions of flow graphs (Diestel, 2000), we treat this structure as a set $\mathcal{W}$ of nodes, which are assigned a set of representatives of the corresponding segment. Edges connecting two nodes representing segments located in their neighbourhood form a set $\mathcal{D}$. To each edge $d \in \mathcal{D}$, a flow value $k \in \mathcal{K}$ is assigned. Thus such a graph can be expressed as

$$\mathcal{G} = (\mathcal{W}, \mathcal{D}, \mathcal{K}). \tag{9}$$

Definition 10. The image of the workspace P in the transformation $\mathbf{f}_x$ is said to be an *autonomous dynamics map*.

Additionally, we can describe some properties of the system, invoked for instance by (in)admissible faults, by means of functions that describe limit values for the pertinent forced dynamics:

$$\mathbf{B} = s_b(\mathbf{x}, t) = \begin{bmatrix} \mathbf{f}_{u\mathrm{MAX}} = \max\limits_{\mathbf{u}\in\mathcal{U}} \mathbf{f}_u(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{f}_{u\mathrm{MIN}} = \min\limits_{\mathbf{u}\in\mathcal{U}} \mathbf{f}_u(\mathbf{x}, \mathbf{u}, t) \end{bmatrix}. \tag{10}$$

Our objective is to find an optimal control $\mathbf{u}^*(t) \in \mathcal{U}$ along with the corresponding state trajectory $E^* = E(\mathbf{u}^*(t)) \in (P \setminus Z)$ which transmits the dynamical system (1) from its initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ to a specified target state $\mathbf{x}(T) = \mathbf{x}_k$ and, at the same time, minimizes the cost functional

$$J(E(\mathbf{u})) = \int_0^T \Big(\sum_{i=1}^{m} \beta_i |\mathbf{u}_i(t)| \Big) \, \mathrm{d}t, \tag{11}$$

where β_i, $i = 1, \ldots, m$ are nonnegative weights and T is a transition time interval resulting from the (optimal) control procedure applied.

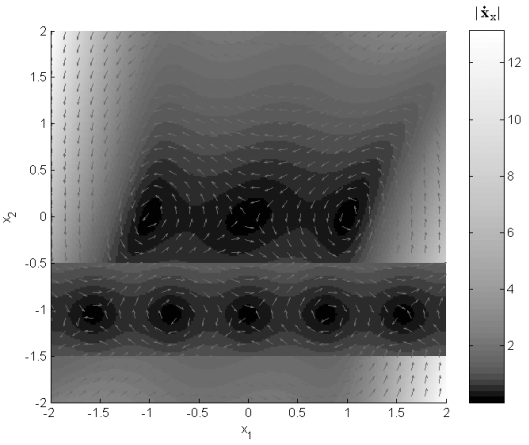

Fig. 1. Autonomous dynamics map.

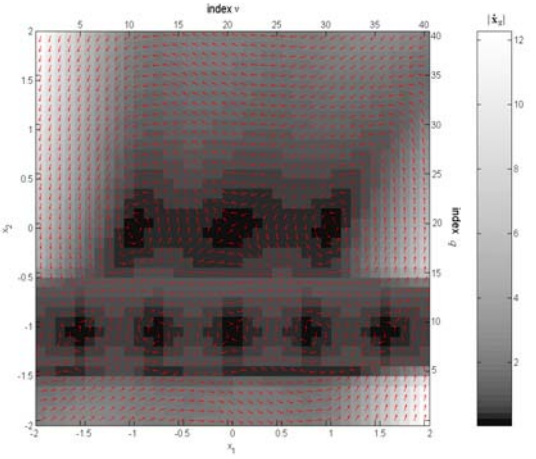

Fig. 2. Segmentation of the autonomous dynamics map.

3. Algorithm description

It is necessary to define four principal and two optional elements: an operational subset P, a transition vector Λ, a cost function $J(E(\mathbf{u}))$, and a segmentation form. Optionally, we can determine a feasible control set $\mathcal{U}$ and a forbidden zone Z. The latter can be used to express the constraints portraying inadmissible faults. We start by segmenting the operational workspace P into a set of N_s segments. Next, for each segment Φ_j, $j = 1, \ldots, N_s$, a set of representatives is determined.

Example 1. Consider a piece of the autonomous dynamics map presented in Fig. 1. Assume that the operational workspace P is represented by a rectangle defined by the vertex pairs $\{(-2,-2), (2,2)\}$. This workspace P is divided into a set of 1600 segments of diameters $d = (0.1 \times 0.1)$, which yields 40 segments per one dimension: $v = 1, \ldots, 40$, $q = 1, \ldots, 40$. The result of this segmentation procedure is depicted in Fig. 2. ♦

Next, we form a state-space graph. Each segment of the workspace P is associated with a certain node (a member of $\mathcal{W}$). An edge connects only those two nodes which represent two segments in their neighbourhood. The flow values are determined according to the cost function and restrictions described below.

An edge $w_a \rightarrow w_b$ is removed from the state-space graph, or equivalently an ∞ value is assigned to it, if for a pair of points $x_1, x_2 : \; x_1 \in \Phi_a, \; x_2 \in \Phi_b$ there is no feasible control $\mathbf{u}(t)$, which could move the operational point from x_1 to x_2 in a finite time.

After such a preparation, we can use any graph search algorithm, which is capable of finding the cheapest path between the initial and terminal nodes, which represents the segments containing the initial and terminal points of the sought trajectory, respectively.

Example 2. With reference to Example 1, let us assign a segment average of the i-th coordinate of the system dynamics as the i-th representative element, $i = 1, \ldots, n$, to the corresponding node of the state-space graph for $j = 1, \ldots, N_s$, with $N_s = 1600$ denoting the number of all segments (or nodes):

$$F_{\mathrm{AVR}\,\Phi_j i} = \frac{1}{N} \sum_{y=1}^{N} F_{\mathrm{DYN}^{i,y}}, \tag{12}$$

where $F_{\mathrm{DYN}^{i,y}}$ stands for the y-th sample of the i-th coordinate of the autonomous dynamics map encoded in $\mathbf{f}_x$, and N denotes the number of samples per segment. Let us assume that $N = 5 \times 5 = 25$, which yields five samples placed each $\Delta_k = (0.1/5) = 0.02$ along the respective axis of the state space. Our objective is to find, for the initial state $\mathbf{x}(0) = \mathbf{x}_0 = [1.9 \; -1.9]^T$ and the terminal state $\mathbf{x}(T) = \mathbf{x}_k = [-1.9 \; 1.9]^T$, an optimal control $\mathbf{u}^*(t) \in \mathcal{U}$, along with its corresponding state trajectory $E^* = E(\mathbf{u}^*(t)) \in \mathcal{X}$ minimizing the cost functional

$$J(E(\mathbf{u})) = \int_0^T \Big(\sum_{i=1}^{m} |u_i(t)| \Big) \, \mathrm{d}t, \tag{13}$$

where T is a transition time related to the duration of the optimal control, and $\mathcal{U}$ and $\mathcal{X}$ denote feasible vector sequences of control and state signals, respectively. In the examples considered, the integration was performed in discrete time and the flow values were calculated according to (13).

In this excercise, Dijkstra's algorithm (for a full description see (Skiena, 1997)), which is suitable for finding the cheapest path in graphs, was effectively utilized for optimization.

Based on the resulting sequence of state-space segments, an approximate continuous reference trajectory was formed by piece-wise linear interpolation based on the centres of the consecutive segments indicated by the optimal path. The results are depicted in Fig. 3. ♦

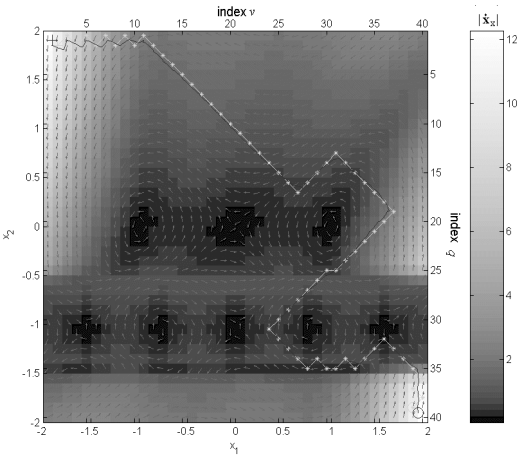

Fig. 3. Segments of an autonomous dynamics map (background), the reference trajectory (-∗-), the initial point (o), the final point (+).

As the forbidden zones are used to indicate which nodes should be excluded from the state-space graph, all the nodes that represent the segments which are partially or entirely included in the defined forbidden zone Z are simply removed from the structure of the state-space graph.

Example 3. Let us extend the above design example by introducing the forbidden zones in the form of the rectangles defined by the following vertex pairs: $\{(0.5,-1.5),(1.5,-0.5)\}$ and $\{(-1,1),(-0.2,1.5)\}$. The resulting state-space trajectories are depicted in Fig. 4. ♦

4. Example of a three tank water system

Consider the system presented in Fig. 5, where three liquid tanks of a height of 5 m are connected through valves A, B and C. Each tank is also supplied with a sink valve controlled by the signals u_1, u_2, u_3, respectively. The valves (A, B, C) are two-stage ON-OFF elements ($u_A, u_B, u_C \in \{0,1\}$) characterized by the openings (slots) $u_A d_{\max}, u_B d_{\max}, u_C d_{\max} \in \{0, 10^{-5}\}$ [m^2], respectively. Moreover, the valves A, B and C have certain dead zones, which means that they can be opened only when the absolute value of the difference between the liquid levels of the adjacent tanks is greater than a given threshold value ϱ [m].

Let us assume the occurrence of two faults: Fault 1, representing a leak in the pipe connecting the valve A with Tank 2, and Fault 2, concerning a leakage located h_{dmg}

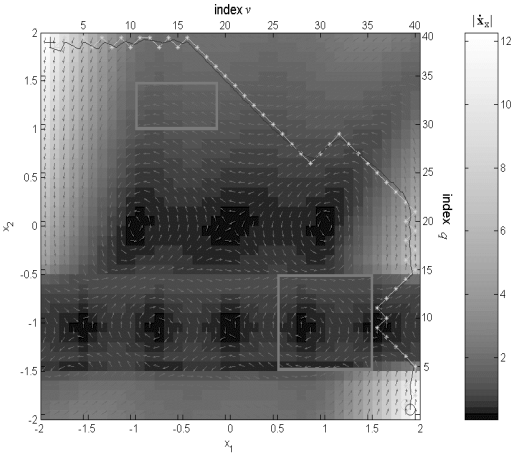

Fig. 4. Segments of an autonomous dynamics map (background), the reference trajectory (-∗-), the initial point (o), the final point (+), the forbidden zones (rectangles) and the control effect (–).

meters above the bottom of Tank 2. The model of this system has the following form:

- state vector $\mathbf{x} = [h_1, h_2, h_3, u_A, u_B, u_C]$, with $h_1, h_2, h_3 \in [0,5]$, $u_A, u_B, u_C \in \{0,1\}$ (the reason for treating u_A, u_B, u_C as states is explained in Section 4.2),
- input vector $\mathbf{u} = [u_p, u_1, u_2, u_3]$, with $u_p \in \{0,1\}; u_1, u_2, u_3 \in [0,1]$,
- differential equations:

$$\begin{aligned}
\frac{\mathrm{d}h_1(t)}{\mathrm{d}t} &= \frac{1}{S}(u_p(t)u_{p\max} - q_1(t) - q_A(t) + q_C(t)),\\
\frac{\mathrm{d}h_2(t)}{\mathrm{d}t} &= \frac{1}{S}(-q_2(t) + q_A(t) - q_B(t)),\\
\frac{\mathrm{d}h_3(t)}{\mathrm{d}t} &= \frac{1}{S}(-q_3(t) + q_B(t) - q_C(t)),
\end{aligned} \tag{14}$$

where

$$\begin{aligned}
q_1 &= u_1(t) d_{\max}\sqrt{2gh_1(t)},\\
q_2 &= u_2(t) d_{\max}\sqrt{2gh_2(t)},\\
q_3 &= u_3(t) d_{\max}\sqrt{2gh_3(t)},
\end{aligned}$$

$$q_A(t) = \begin{cases}
(1-\eta)u_A(t)d_{\max}\sqrt{2gh(h_1(t)-h_2(t))} \\
\qquad \text{for } h_1(t) \geq h_2(t) + \varrho_A, \\
(1-\eta)u_A(t)d_{\max}\sqrt{2gh(h_2(t)-h_1(t))} \\
\qquad \text{for } h_2(t) \geq h_1(t) + \varrho_A, \\
0 \qquad \text{otherwise},
\end{cases}$$

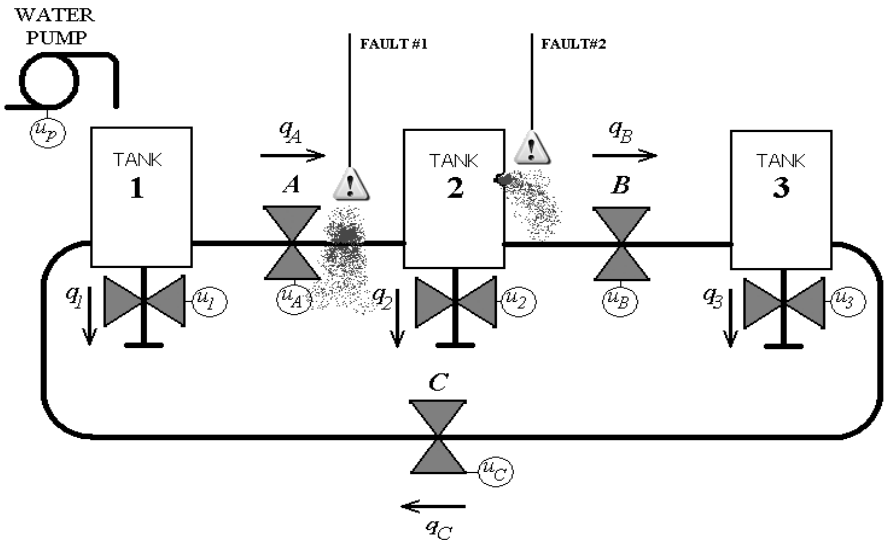

Fig. 5. Three-tank system with two leakages.

$$q_B(t) = \begin{cases} u_B(t) d_{\max} \sqrt{2gh(h_2(t) - h_3(t))} \\ \qquad \text{for } h_2(t) \geq h_3(t) + \varrho_B, \\ u_B(t) d_{\max} \sqrt{2gh(h_3(t) - h_2(t))} \\ \qquad \text{for } h_3(t) \geq h_2(t) + \varrho_B, \\ 0 \qquad \text{otherwise,} \end{cases}$$

$$q_C(t) = \begin{cases} u_C(t) d_{\max} \sqrt{2gh(h_3(t) - h_1(t))} \\ \qquad \text{for } h_3(t) \geq h_1(t) + \varrho_C, \\ u_C(t) d_{\max} \sqrt{2gh(h_1(t) - h_3(t))} \\ \qquad \text{for } h_1(t) \geq h_3(t) + \varrho_C, \\ 0 \qquad \text{otherwise,} \end{cases}$$

• constants:
$d_{\max} = 10^{-5}\text{m}^2$ – the maximal opening of the valve,
$u_{p\max} = 0.2 \cdot 10^{-4}\text{m}^3/s$ – the maximal pump output,
$S = 0.1\,\text{m}^2$ – the area of the tank bottom,
$\eta \in [0, 1]$ – the weight of Fault 1.

As a result of the decomposition (14) into the autonomous and forced dynamics, we have

$$\dot{\mathbf{x}}_u = \begin{bmatrix} \frac{1}{S}(u_p(t) u_{p\max} - q_1(t)) \\ \frac{1}{S}(-q_2(t)) \\ \frac{1}{S}(-q_3(t)) \end{bmatrix}, \qquad (15)$$

$$\dot{\mathbf{x}}_x = \begin{bmatrix} \frac{1}{S}(-q_A(t) + q_C(t)) \\ \frac{1}{S}(q_A(t) - q_B(t)) \\ \frac{1}{S}(q_B(t) - q_C(t)) \end{bmatrix}. \qquad (16)$$

The limit values for the forced dynamics are defined as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{f}_{u\text{MAX}} \\ \mathbf{f}_{u\text{MIN}} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} u_{p\max} \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} -d_{\max}\sqrt{2gh_1} \\ -d_{\max}\sqrt{2gh_2} \\ -d_{\max}\sqrt{2gh_3} \end{bmatrix} \end{bmatrix}. \qquad (17)$$

4.1. Problem formulation. Consider the problem of finding optimal trajectories while minimizing the cost function

$$J_k(\mathbf{x}_0, \mathbf{x}_k) = \int_{x_0}^{x_k} |u(\mathbf{x})|\, \mathrm{d}\mathbf{x}, \qquad (18)$$

where x_0 and x_k are the points defined by the transition vector Λ. Let us assume that $\varrho_A = \varrho_B = 0.5$ m and $\varrho_C = 3.5$ m.

4.2. Sketch of implementation.
Operational space. The model (14) can be described by the state vector $[h_1\ h_2\ h_3\ u_A\ u_B\ u_C]^T$ from a six-dimensional space. Note that (during this off-line stage) the controlling valve inputs u_A, u_B, u_C are included in the state in order to plan the open-loop control of the valves A,

B and C. By using this information and taking into consideration the height of the liquid tanks, the operational workspace can be defined as $P = \{h_1, h_2, h_3, u_A, u_B, u_C : h_1, h_2, h_3 \in [0,5], u_A, u_B, u_C \in \{0,1\}\}$.

Segmentation. Let us fix the size of the segments as [0.25 m×0.25 m × 0.25 m ×{0, 1} × {0, 1} × {0, 1}]. This yields $20^3 \cdot 2^3 = 64000$ segments $\Phi_{v_1,v_2,v_3,q_1,q_2,q_3}$ in P, indexed by $v_1, v_2, v_3 = 1, \ldots, 20$, and $q_1, q_2, q_3 = 1, 2$.

In general, any two segments sharing at least one point in the state space are treated as neighbours. In the case considered we shall confine the notion of the neighbourhood by saying that two segments which have a common side-wall, an edge or a vertex in the state space of the levels $(h_1 \times h_2 \times h_3)$ are neighbours. To comply with these demands, we use the following elements as the representatives; for each segment Φ_j, $j = 1, \ldots, N_s$, where $N_s = 64000$ and $i = 1, \ldots, 3$; $y = 1, \ldots, 343$:

1. The average of the segment dynamics in a particular direction

$$F_{\mathrm{AVR}\,\Phi_j i} = \frac{1}{N} \sum_{y=1}^{N} F_{\mathrm{DYN}^{i,y}},$$

where $F_{\mathrm{DYN}^{i,y}}$ stands for the y-th sample of the i-th coordinate of the autonomous dynamics map encoded in $\mathbf{f}_x$, and N denotes the number of samples per segment, $N = (7{\times}7{\times}7{\times}1{\times}1{\times}1) = 343$ (along $h_1, h_2, h_3, u_A, u_B, u_C$), taken at a sample step $\Delta_k = (0.25/7) \simeq 0.0357$.

2. A maximal value of the i-th map component

$$M_{\Phi_j,i} = \max_y (F_{\mathrm{DYN}^{i,y}}). \tag{19}$$

3. A minimal value of the i-th map component

$$W_{\Phi_j,i} = \min_y (F_{\mathrm{DYN}^{i,y}}). \tag{20}$$

4. An upper limit of the i-th forced dynamics

$$U^+_{\Phi_j,i} = \min_y (f_{u_j \mathrm{MAX}^y}). \tag{21}$$

5. A lower limit of the i-th forced dynamics

$$U^-_{\Phi_j,i} = \max_y (f_{u_j \mathrm{MIN}^y}). \tag{22}$$

6. An index vector $[v_1\ v_2\ v_3\ q_1\ q_2\ q_3]^T$.

The cost of the path between the initial node and the current one, where $\mathbf{x}_0, \ldots, \mathbf{x}_k$ represent the geometrical centres of the segments in the operational workspace, is $c = J(\mathbf{x}_0, \mathbf{x}_1) + J(\mathbf{x}_1, \mathbf{x}_2) + \cdots + J(\mathbf{x}_{k-1}, \mathbf{x}_k)$.

For each edge $d_{O,S} : w_O \rightarrow w_S$ for $O, S = 1, \ldots, N_s$ and $O \neq S$, which connects the nodes that represent the segments Φ_O and Φ_S, we calculate the flow value $k_{w_O \rightarrow w_S}$, considering only the workspace of the levels h_1, h_2, h_3 in the state space (since alternating the states of the valves A, B and C is always possible and is not represented in the assumed control cost indicator).

In order to calculate the flow cost $k_{w_O \rightarrow w_S}$ assigned to the edge $d_{O,S} : w_O \rightarrow w_S$, we consider the three level dimensions ($i = 1, \ldots, 3$: h_1, h_2, h_3) separately. Thus a total cost was calculated as the respective sum $\sum_{i=1}^{3} [k_{w_O \rightarrow w_S}]_{\Phi_O,i} + \sum_{i=1}^{3} [k_{w_O \rightarrow w_S}]_{\Phi_S,i}$, where $[k_{w_O \rightarrow w_S}]_{\Phi_O,i}$ is a partial cost calculated for the segment Φ_O and $[k_{w_O \rightarrow w_S}]_{\Phi_S,i}$ is the cost calculated for Φ_S. Let us denote $I_e = I_S - I_O = [v_{e_1}\ v_{e_2}\ v_{e_3}]^T = [v_{S_1}\ v_{S_2}\ v_{S_3}]^T - [v_{O_1}\ v_{O_2}\ v_{O_3}]^T$, where I_O, I_S are the indices of Φ_O and Φ_S, respectively. For Φ_O, these calculations were performed in the following form:

FOR $i = 1, 2$

- Case $I_{ei} = 1$:

IF sign(I_{ei}) = sign$(F_{\mathrm{AVR}\Phi_O,i})$ THEN:
$[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = 0$
(*the i-th component of the autonomous dynamics is suitable for the transition between the segments represented by the nodes w_O, w_S*)
ELSE: (*it is necessary to check if the feasible values of the forced dynamics enable moving the operational point beyond the boundary of the segment Φ_O in the desired direction*)
IF $W_{\Phi_O,i} + U^+_{\Phi_O,i} > 0$ THEN:
$[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \alpha(|F_{\mathrm{AVR}\Phi_O,i}| + \epsilon)$ (*the cost is proportional to the value of the autonomous dynamics and some positive constant*)
ELSE: $[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \infty$.

- Case $I_{ei} = 0$:

IF $W_{\Phi_O,i} + U^+_{\Phi_O,i} \geq 0$
$\wedge M_{\Phi_O,i} + U^-_{\Phi_O,i} \leq 0$ THEN:
$[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \alpha |F_{\mathrm{AVR}} \Phi_O, i|$
ELSE: $[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \infty$.

- Case $I_{ei} = -1$ (*see the three remarks above*):

IF sign(I_{ei}) = sign$(F_{\mathrm{AVR}\Phi_O,i})$ THEN:
$[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = 0$
ELSE:
IF $M_{\Phi_O,i} + U^-_{\Phi_O,i} < 0$ THEN:
$[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \alpha(|F_{AVR\Phi_O,i}| + \epsilon)$
ELSE: $[k_{w_O \rightarrow w_S}]_{\Phi_O,i} = \infty$.

The analysis for the segment Φ_S was performed in a similar way.

The coefficient α is a normalized distance between the geometrical centres of the segments Φ_O and Φ_S. If the segments Φ_O and Φ_S have a common side wall in the level space $h_1 \times h_2 \times h_3$, then $\alpha = 1$. If the segments Φ_0 and Φ_S have one common edge in this space, then $\alpha = \sqrt{2}$. If they have one common node/point, then $\alpha = \sqrt{3}/2$.

In the case of $I_e = 0$, the compensation of the autonomous dynamics is sufficient, while in the case when one of the coordinate directions is nonzero ($I_{ei} = 1$ or -1), an additional control effort is necessary, which is portrayed in the control cost by ϵ.

As before, the first off-line stage of this suboptimal procedure consisted in seeking an optimal path within the state-space graph. This task was performed by using Dijkstra's algorithm (Skiena, 1997). We assume that the geometrical centres of these segments form a list of reference points. Such a reference trajectory can be tracked by an executive proportional or predictive controller (see, e.g., (Kowalczuk and Suchomski, 2005)) in an on-line procedure of suboptimal fulfillment of the control task.

As an illustrative example, we consider the above-described system working in the presence of two faults. Our goal is to fill up Tanks 2 and 3 in a way that the total amount of liquid stored in those two tanks is equal to $0.197\,\mathrm{m}^3$. We treat Fault 1 as an admissible one and Fault 2 as an inadmissible one. Note that such an approach is nowadays practiced under the concept of reconfigurable control, based on fault detection and diagnosis (Zhang, 2007).

Taking into consideration the first fault, and assuming its weight as $\eta = 0.34$, we fix the equations (14). The leakage from Tank 2, represented by Fault 2, occurs when the level of the liquid is equal to or greater than $h_{dmg} = 1.125$ m. We deal with this problem by defining an additional constraint of the following form:

$$h_2 < h_{dmg}. \tag{23}$$

Taking account of this constraint, we simply introduce the forbidden zone Z in the state space as a set of the states satisfying $Z = \{\mathbf{x}_z = (h_1, h_2, h_3, u_A, u_B, u_C) : h_2 \geq h_{dmg}\}$. By doing this we are able to exclude all the nodes from the state-space graph that represent the segments which partially or entirely belong to the forbidden zone Z. The initial node was assumed as $w_{1,1,1,1,1,1}$, and the terminal node was set to any nonexcluded node representing a segment, located above by the line $h_2+h_3 = 1.97$ m. The obtained results are illustrated in Fig. 6.

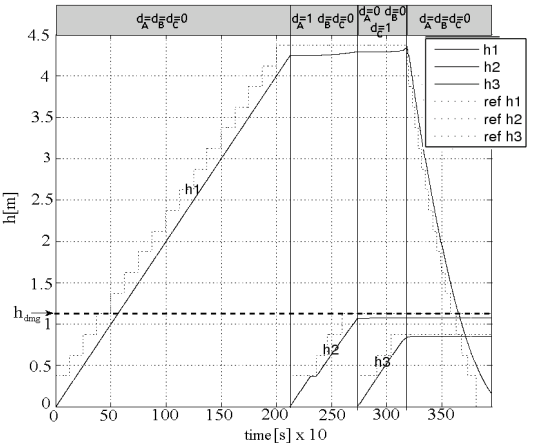

Fig. 6. On-line control effects for the considered three-tank problem: solid lines – the levels gained by the executive controller, dotted lines – the reference trajectories.

5. Summary

This work presents a useful concept of planning the optimal and safe control strategies based on discrete optimization algorithms.

Certainly, there are shortcomings of this method associated with the general nature of graph representations. Namely, in the case of high-order models we can easily be faced with the problem of complexity resulting from a high number of nodes characterizing the related state-space graph. This, eventually, may lead to time-consuming graph search procedures. Some clues concerning the possibilities of improving the graph search process can be found, e.g., in (Bertsekas, 2005; Friedman and Tarjan, 1987).

Taking into account the fact that the reference trajectories are portrayed in the piece-wise linear form based on a given rough segmentation scheme, some problems may occur in cases when the optimal trajectory of the operational point needs a smooth representation.

Similarly, in spite of the optimality in the sense of the graph representation, taking into consideration the original design problem, we shall be aware of the suboptimal property of this approach resulting from the fact that the initial and terminal points of the trajectory may not be identical with the centres of the initial and terminal segments, respectively. Thus, in general, as the reference trajectory can only pass through the centres of the segments, it is clear that the size of segments has great influence on the quality of the ultimate optimal solution found.

References

Bertsekas D. P. (2005). *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA.

Chowdhury F. N. and Chen W. (2006). Fault monitoring in the presence of fault-tolerant control, *Proceedings of the 6th IFAC Symposium on SAFEPROCESS: Fault Detection, Supervision and Safety of Technical Processes*, Vol. 1, IFAC, Beijing, China, pp. 1321–1326.

Diestel R. (2000). *Graph Theory*, Springer-Verlag, New York, NY.

Friedman M. L. and Tarjan R. E. (1987). Fibbonaci heaps and their uses in improved network optimization algorithms, *Journal of the Association for Computing Machinery* **34**(3): 596–615.

Kowalczuk Z. and Olinski K. E. (2007). Sub-optimal fault-tolerant control with the use of discrete optimization, *in* J. Korbicz, K. Patan and M. Kowal (Eds.), *Fault Diagnosis and Fault Tolerant Control*, Academic Publishing House EXIT, Warsaw, pp. 165–172.

Kowalczuk Z., Rudzinska-Kormanska K. and Olinski K. E. (2007). Designing nonlinear control systems by state-space flow graph optimization, *Proceedings of the 11th IFAC Symposium on Large Scale Systems*, Gdańsk, Poland, pp. 1–4, (on CD-ROM).

Kowalczuk Z. and Suchomski P. (2005). Discrete-time predictive control design based on overparameterized delay-plant models and identified cancellation order, *International Journal of Applied Mathematics and Computer Science* **15**(1): 5–34.

Skiena S. S. (1997). *The Algorithm Design Manual*, Springer-Verlag, New York, NY.

Zhang Y. (2007). Active fault-tolerant control systems: Integration of fault diagnosis and reconfigurable control, *in* J. Korbicz, K. Patan and M. Kowal (Eds.), *Fault Diagnosis and Fault Tolerant Control*, Academic Publishing House EXIT, Warsaw, pp. 21–41.

Received: 18 December 2007
Revised: 12 March 2008