INFINITE HORIZON INPUT SIGNAL FOR ACTIVE FAULT DETECTION IN CONTROLLED MARKOV CHAINS

Ivo Punčochář and Miroslav Šimandl
Department Cybernetics and European Centre of Excellence – NTIS
University of West Bohemia
Univerzitní 8
Pilsen, Czech Republic
emails: ivop@kky.zcu.cz, simandl@kky.zcu.cz

ABSTRACT
The paper deals with the problem of active fault detection with a given fault detector over an infinite time horizon. Systems that can be modeled using two interconnected discrete-time finite-state Markov chains are considered. The first Markov chain is unobservable and describes switching between fault-free and faulty modes. The second one is an observable controlled Markov chain that describes the system dynamics in fault-free and faulty modes. The maximum a posteriori probability fault detector is assumed, and an input signal generator that improves the decision quality is designed. The original problem is reformulated as a perfect state information problem and solved by dynamic programming. An infinite time horizon is considered to reduce off-line computational demands and a perceptron neural network is employed to lower memory requirements for on-line use. The results are illustrated in a numerical example.

KEY WORDS
Active fault detection, Controlled Markov chains, Input signal design

1 Introduction
Fault detection deals with discerning certain changes in an observed system. These changes are considered to be faults that need to be detected and accommodated in order to prevent failures with harmful consequences such as increased costs, health threats and other [1, 2].

There are two approaches to fault detection called passive and active. A typical block diagram of passive fault detection for a closed loop system is depicted in Fig. 1. A passive fault detector uses the a priori information, inputs \( u \), and outputs \( y \) to generate decisions \( d \) whether a fault has occurred in the observed system or not [3]. Although the passive fault detectors provide a satisfactory quality of detection in most cases, there are situations in which the input-output data generated within the closed loop system are not informative enough for the passive fault detector to recognize a fault reliably and in time.

To enable or improve fault detection in such situations, the active approach to fault detection has been intro-

Figure 1. Typical block diagram of passive fault detection

duced. This approach originates in optimum experimental designs [4, 5] and optimal input signal design for parameter estimation [6]. The goal is to design an active fault detector that besides decisions generates an auxiliary input signal that is fed back into the system to facilitate fault detection. Such a problem has been considered for stochastic models [7, 8] and later on also for deterministic uncertain models [9, 10].

Since all inputs of the system are usually used to close the control loop, the active approach to fault detection leads inevitably to the problem of active fault detection and control (AFDC). A block diagram of active fault detection and control is depicted in Fig. 2, where an active fault detector and controller processes the output \( y \) of the system and generates the decision \( d \) and input \( u \). The AFDC problem was considered for deterministic uncertain models e.g. in [11]. For stochastic models, the AFDC problem was treated in [12], and a unified formulation of AFDC on a finite time horizon for discrete-time stochastic models together with a formal solution based on the closed loop (CL) information processing strategy (IPS) was studied in [13, 14]. The unified formulation provides a theoretical framework that allows several special cases and their mutual relationships to be analyzed.

The design of an optimal active fault detector and controller based on the CL ISP requires to solve a functional equation for which an analytical solution is hard to find and a numerical solution is intractable. Thus, an approximation of the optimal closed loop solution [15] or use of another IPS [16] were considered. In both cases substantial on-line computation is required to keep the advantages of the CL IPS. In the case of active robust fault detection for uncer-
tain deterministic models, the computational issues were addressed using the asymptotic infinite time horizon solution [17]. This paper addresses a similar issue for a special case of active fault detection in stochastic models.

This paper focuses on a special case of AFDC, where an input signal generator is to be designed for a given fault detector [14]. The problem is formulated over an infinite time horizon for discrete-time finite-state controlled Markov chains with partially observed state using a discounted criterion. An input signal generator that requires little on-line computation and can be used as an approximation for the finite time horizon problems is designed.

The paper is organized as follows. The considered problem is formulated in Section 2. The original problem is transformed into a perfect state information problem in Section 3 by introducing a new augmented state. The design of the optimal and approximate input signal generator is presented in Section 4. The proposed approach is demonstrated in a numerical example in Section 5 and final Section 6 draws some conclusions.

2 Problem formulation

The block diagram of active fault detection with a given detector is depicted in Fig. 3. The description of the observed system, given fault detector, input signal generator, and criterion is given in this section.

2.1 Observed system

Let us assume that the observed system can be described using the following model

\[ P^x(x_{k+1}|x_k, \mu_k, u_k), \quad k = 0, 1, \ldots \]  \hspace{1cm} (1a)

\[ P^\mu(\mu_{k+1}|\mu_k), \quad k = 0, 1, \ldots \]  \hspace{1cm} (1b)

where the couple \([x_k, \mu_k]\) denotes the state of the observed system and \(u_k \in \mathcal{U} = \{1, 2, \ldots, n_u\}\) is the input. The part of the state \(x_k \in \mathcal{X} = \{1, 2, \ldots, n_x\}\) is a directly observed controlled Markov chain, whereas the other part of the state \(\mu_k \in \mathcal{M} = \{1, 2\}\) is a Markov chain that cannot be observed directly. The first mode \(\mu_k = 1\) represents a fault-free behavior of the observed system and the second mode \(\mu_k = 2\) represents a faulty behavior. The only information regarding the mode \(\mu_k\) can be inferred from the input \(u_k\) and observed part of the state \(x_k\). Besides known transition probability distributions \(P^x\) and \(P^\mu\) also the probability distributions of initial conditions \(P^x(x_0)\) and \(P^\mu(\mu_0)\) are given.

Note that although the model (1) might seem quite simplistic at first, the controlled Markov chain (1a) can represent an approximate model of a complex discrete-time nonlinear stochastic system whose state space is quantized to obtain a finite-state representation with corresponding transition probabilities [18, 19].

2.2 Given fault detector

It is assumed that the following fault detector is given

\[ d_k = \sigma_k(I_k) = \arg \max_{j \in \mathcal{M}} P(\mu_k = j|I_k), \]  \hspace{1cm} (2)

where \(d_k \in \mathcal{M}\) is a decision whether the system is in the fault-free or faulty mode, \(I_k = [x_k^0, u_k^{k-1}]\) is an information vector, \(\sigma_k : \mathcal{X} \times \mathcal{U}^k \rightarrow \mathcal{M}\) is a given function that describes the fault detector, and \(P(\mu_k|I_k)\) is the conditional probability of the mode \(\mu_k\). The decision \(d_k\) can be viewed as the point estimate of the mode \(\mu_k\), which is optimal in the maximum a posteriori probability sense. Note that the given fault detector is a time-varying system and the dimension of its argument is expanding in time.

The conditional probability \(P(\mu_k = j|I_k)\) can be computed recursively according to

\[ P(\mu_k = j|I_k) = \sum_{i=1}^{2} P_{ij}^\mu P(\mu_{k-1} = i|I_k), \]  \hspace{1cm} (3)

where \(P_{ij}^\mu = P^\mu(\mu_{k+1} = j|\mu_k = i)\) and the conditional probability \(P(\mu_{k-1} = i|I_k)\) is given as

\[ P(\mu_{k-1} = i|I_k) = \frac{P^x(x_{k-1}|x_{k-1}, \mu_{k-1} = i, u_{k-1})P(\mu_{k-1} = i|I_{k-1})}{P(x_k|I_{k-1}, u_{k-1})}. \]  \hspace{1cm} (4)
Finally, the predictive probability $P(x_k|I_{k-1}, u_{k-1})$ is computed according to

$$P(x_k|I_{k-1}, u_{k-1}) = \sum_{j=1}^{2} P^x(x_k|x_{k-1}μ_{k-1} = j, u_{k-1}) \times P(μ_{k-1} = j|I_{k-1}).$$

(5)

The recursive computation is initialized with the known probability distribution $P(μ_0|x_0) = P(μ_0)$.

### 2.3 Input signal generator

It is assumed that the input signal generator can generally be described in the following way

$$u_k = γ_k(I_k),$$

(6)

where $γ_k : \mathcal{X}^{k+1} × \mathcal{U}^k \mapsto \mathcal{U}$ is an unknown function. The sequence of functions $γ_k$ for $k = 0, 1, \ldots$ is called policy $π = \{γ_0, γ_1, \ldots\}$. The goal is to select an admissible policy $π$ such that the quality of detection measured by a chosen criterion is optimized.

### 2.4 Criterion

The quality of individual policies $π$ is evaluated using the following additive criterion

$$J(π) = \lim_{F \to \infty} E \left\{ \sum_{k=0}^{F} L_k^*(μ_k, d_k) \right\},$$

(7)

where $F$ is the final time step of the considered horizon, $L_k^* : \mathcal{M} × \mathcal{M} \mapsto \mathbb{R}^+$ is a detection cost function, and $E\{\cdot\}$ is the expectation operator over all included random variables. To make the criterion (7) well-defined and bounded for all admissible policies, the discounted detection cost function $L_k^*$ of the following form is used

$$L_k^*(μ_k, d_k) = \lambda^k L^d(μ_k, d_k),$$

(8)

where $\lambda \in (0, 1)$ is a discount factor and $L^d : \mathcal{M} × \mathcal{M} \mapsto \mathbb{R}^+$ is chosen to be the zero-one cost function

$$L^d(μ_k, d_k) = \begin{cases} 0 & d_k = μ_k, \\ 1 & \text{otherwise}, \end{cases}$$

(9)

that penalizes incorrect decisions. Although the discount factor $\lambda$ is primarily included into the criterion to make it bounded, it can also be viewed as a way of expressing the fact that the quality of decisions in far future is less important than the quality of immediate decisions.

### 3 Reformulation using belief states

The presented problem belongs among the imperfect state information problems. To illustrate its several properties, a finite time horizon solution is derived and discussed first. Then a standard approach to handle imperfect state information problems on an infinite time horizon is applied. The original problem is reformulated as a perfect state information problem using a belief state and solved using dynamic programming [20].

#### 3.1 Optimal finite horizon active fault detection with given detector

First, the optimal solution to the finite time horizon problem is briefly presented, see e.g. [13] for details. If it is assumed that the final time step $F$ is optimal, the optimal input signal generator can be obtained by solving the following backward recursive equation for $k = F, F-1, \ldots, 1, 0$

$$V^*_k(I_k) = \min_{u_k \in \mathcal{U}} \left\{ L_k^d(μ_k, d_k) + V^*_{k+1}(I_{k+1}|I_k, u_k, d_k) \right\},$$

(10)

where $E\{\cdot|\cdot\}$ is the conditional expectation operator, decision $d_k$ is specified by the given fault detector (2), and $V^*_k : \mathcal{X}^{k+1} × \mathcal{U}^k \mapsto \mathbb{R}$ is a function, called the cost-to-go function [21] or the optimal value function [22], that represents the minimum expected costs incurred during time steps $k$ through $F$ conditioned by the information vector $I_k$.

The initial condition for the backward recursive equation is $V_{F+1} = 0$ for all $I_{F+1}$, and the minimum value of the criterion is given as $J(γ^{opt}_0) = E[V^*_0(I_0)]$.

The optimal input signal generator is given as

$$u_k^* = γ_k^*(I_k) = \arg \min_{u_k \in \mathcal{U}} Q_k^*(I_k, u_k)$$

(11)

where $Q_k^* : \mathcal{X}^{k+1} × \mathcal{U}^{k+1} \mapsto \mathbb{R}$ was introduced mainly to simplify notation, but it can also be viewed as an analog to the Q-function in the case of infinite time horizon problems. The optimal input signal generator is a time-variant system, and it can be seen from Tab. 1 that memory requirements to store the optimal input signal generator increase exponentially as the length of the horizon $F$ grows.

#### Table 1. Memory requirements to represent the optimal input signal generator for the finite horizon $F$

<table>
<thead>
<tr>
<th>Representation by</th>
<th>The number of function values</th>
</tr>
</thead>
<tbody>
<tr>
<td>functions $γ^*_0$</td>
<td>$n_x(n_xn_u)^{F+1-1}n_xn_u^{-1}$</td>
</tr>
<tr>
<td>functions $V^*_0$</td>
<td>$n_x(n_xn_u)^{F+1-1}n_xn_u^{-1}$</td>
</tr>
<tr>
<td>functions $Q^*_0$</td>
<td>$n_xn_u(n_xn_u)^{F+1-1}n_xn_u^{-1}$</td>
</tr>
</tbody>
</table>

The ever expanding dimension of the information vector $I_k$ can be circumvented by using a sufficient statis-
tic [21]. By substituting the detection cost function (8) together with the zero-one cost function (9) into the backward recursive equation (10) and using the definition of the given fault detector (2), the backward recursive equation can be rewritten as

\[ \bar{V}_k^*(I_k) = \min_{u_k \in U} \left[ P(\mu_k = 1|I_k), P(\mu_k = 2|I_k) \right] + \lambda \min_{u_k \in U} \mathbb{E} \left\{ \bar{V}_{k+1}^*(I_{k+1}) | I_k, u_k, d_k \right\}, \]  

where \( \bar{V}_k^*(I_k) = \lambda^k \bar{V}_1^*(I_k) \). By inspection of the backward recursive equation, it can be shown that the coupled \([x_k, P(\mu_k = 1|I_k)]\) is a sufficient statistic for this problem. These partial results will be employed in the next subsection that deals with the infinite time horizon version.

### 3.2 Optimal infinite horizon active fault detection with given detector

Using the result obtained in the previous subsection, the original problem can be reformulated as a perfect state information problem in the following way. Let us define a new state of the model as

\[ s_k = \begin{bmatrix} x_k \\ b_k \end{bmatrix}, \]  

where \( b_k = P(\mu_k = 1|x_0^k, u_0^{k-1}) \in [0, 1] \) is a belief state [20]. Using the new state \( s_k \), the original model can be expressed as

\[ s_{k+1} = \phi(s_k, u_k, x_{k+1}) = \begin{bmatrix} \phi_1(s_k, u_k, x_{k+1}) \\ \phi_2(s_k, u_k, x_{k+1}) \end{bmatrix}, \]  

where \( x_{k+1} \) is now regarded as an external disturbance with the known probability distribution \( P(x_{k+1}|s_k, u_k) \) that is identical with (5). \( \phi_1(s_k, u_k, x_{k+1}) = x_{k+1} \), and \( \phi_2 \) can be expressed using (3)-(5) as

\[ \phi_2(s_k, u_k, x_{k+1}) = P_{11}^u P_x(x_{k+1}|x_k, \mu_k = 1, u_k) b_k + P_{12}^u P_x(x_{k+1}|x_k, \mu_k = 2, u_k) (1 - b_k) / c; \]  

where the normalization constant \( c \) is given by

\[ c = P'_x(x_{k+1}|x_k, \mu_k = 1, u_k) b_k + P'_x(x_{k+1}|x_k, \mu_k = 2, u_k) (1 - b_k). \]  

As \( x_0 \) is directly observed and \( b_0 = P^u(\mu_0 = 1) \) is known, the initial state \( s_0 \) is fully specified.

Using the new state \( s_k \), the given fault detector (2) can be rewritten as a time-invariant system

\[ d_k = \sigma(s_k) = \begin{cases} 1 & \text{if } b_k \geq 1 - b_k, \\ 2 & \text{if } b_k < 1 - b_k. \end{cases} \]  

Since the observed system and given fault detector are now described as time-invariant systems, it is sufficient to look for the optimal input signal generator among the stationary input signal generators of the form

\[ u_k = \gamma(s_k), \]  

where \( \gamma : \mathcal{X} \times [0, 1] \rightarrow U \) is a stationary strategy. Finally, starting at the state \( s_0 \), the criterion can be written as

\[ \bar{J}(\gamma, s_0) = \lim_{F \to \infty} \mathbb{E} \left\{ \sum_{k=0}^{F} \lambda^k \bar{L}^d(s_k) | s_0 \right\}. \]  

Now the original problem is reformulated and specified by the state equation (14), stationary fault detector (17) and criterion (19). This form is more suitable for input signal generator design using a standard techniques of dynamic programming.

### 4 Design of input signal generator

The optimal input signal generator is characterized first and then an approximate solution based on discretization of the belief state \( b_k \) is presented.

#### 4.1 Optimal input signal generator

The optimal input signal generator for an infinite time horizon can be obtained from the optimal value function [23, 22]. Consider a function \( V^*(s_k) \) that represents the value of the criterion (19) when starting at the state \( s_k \) and following a stationary strategy \( \gamma \). By the problem specification, this function is well-defined and bounded for all possible stationary strategies. The optimal value function \( V^*(s_k) = V^\gamma(s_k) \) can be found by solving the following Bellman’s functional equation

\[ V^*(s_k) = \min_{u_k \in U} \left\{ \bar{L}^d(s_k) + \lambda \mathbb{E} \{ V^*(s_{k+1}) | s_k, u_k \} \right\}. \]  

Since the detection cost \( \bar{L}^d \) is independent of the input \( u_k \), the optimal input signal generator \( \gamma^* \) is obtained from the optimal value function as

\[ u^*_k = \gamma^*(s_k) = \arg \min_{u_k \in U} \mathbb{E} \{ V^*(s_{k+1}) | s_k, u_k \}, \]  

or from the optimal Q-function \( Q^*(s_k, u_k) \) as

\[ u^*_k = \gamma^*(s_k) = \arg \min_{u_k \in U} Q^*(s_k, u_k). \]  

If the optimal strategy \( \gamma^* \) is computed and stored, it can be used on-line directly to find the optimal inputs. If the optimal Q-function \( Q^*(s_k, u_k) \) is stored instead, the optimal inputs can be computed on-line by solving a simple minimization problem defined in (23).
4.2 Suboptimal input signal generator

The Bellman’s functional equation (20) poses a complex functional problem that needs to be solved numerically. There are several methods for solving the Bellman’s functional equation, such as the value iteration, policy iteration or linear programming [23]. In this paper an approximation of the optimal value function \( V^* \) will be computed using the value iteration algorithm. The value iteration algorithm computes successive approximations of the optimal value function \( V^* \) using the following functional recursive equation

\[
V^{(i+1)}(s_k) = \min[b_k, 1 - b_k] + \lambda \min_{u_k \in U} \{ V^{(i)}(s_k+1)|s_k, u_k \},
\]

(24)

where \( V^{(i)} : \mathcal{X} \times [0,1] \rightarrow \mathbb{R} \) is a function that approximates the optimal value function \( V^* \) at the \( i \)-th iteration. The choice of the initial approximate value function \( V^{(0)} \) influences the number of iterations required to obtain a desired accuracy. Here, the initial approximate value function is chosen to be \( V^{(0)}(s_k) = 0 \) for all \( s_k \). Considering the model (14), the conditional mean value on the right hand side of (24) can be evaluated as

\[
E \{ V^{(i)}(s_{k+1})|s_k, u_k \} = \sum_{i=1}^{n_s} V^{(i)}(x_k, \phi_2(s_k, u_k, i) P(x_{k+1} = i|x_k, b_k, u_k),
\]

(25)

where the probability \( P(x_{k+1} = i|x_k, b_k, u_k) \) is given by

\[
P^{\pi}(x_{k+1} = i|x_k, b_k, u_k) = \begin{cases} P^{x}(x_{k+1} = i|x_k, \mu_k = 1, u_k) b_k + P^{x}(x_{k+1} = i|x_k, \mu_k = 2, u_k)(1 - b_k). 
\end{cases}
\]

The iteration step performed in the value iteration algorithm is tightly related to the backward recursive equation (10). It gives ground for considering the infinite time horizon solution as an approximation to the finite time horizon equivalent problem. This behavior is described by the so called turnpike theorems [22].

The approximate value function \( V^{(i)} \) approaches the optimal value function \( V^* \) asymptotically as \( i \rightarrow \infty \) [23], and the norm the approximation error after a finite number of iterations \( i \) is bounded as

\[
||V^{(i)}(s_k) - V^*(s_k)||_{\infty} \leq \lambda^i ||V^*(s_k)||_{\infty}.
\]

(27)

In practice a finite number of iteration is used with a stopping criterion [23]. It is a common practise to terminate the value iteration algorithm when

\[
||V^{(i)}(s_k) - V^{(i-1)}(s_k)||_{\infty} \leq \delta(1 - \lambda),
\]

(28)

where \( \delta \) is a chosen upper bound on the approximation error, i.e. \( ||V^{(i)}(s_k) - V^*(s_k)||_{\infty} \leq \delta \). The final iteration is further denoted \( i_{\text{final}} \).

Contrary to the state variable \( x_k \) and the input \( u_k \) that have discrete domains by the problem specification, the belief state \( b_k \) has a continuous state space. Therefore an approximate representation of the functions \( V^{(i)}(x_k, b_k) \) is needed. It can be shown that for a fixed \( x_k = \bar{x} \in \mathcal{X} \) the functions \( V^{(i)}(x_k, b_k) \) are piecewise linear concave functions of the belief state \( b_k \in [0,1] \) [21]. To avoid a complex computation of the exact value function, it is approximated instead. First, for the value iteration algorithm that is performed off-line, the value function is approximated using a uniform grid of points over the state space of the belief state \( b_k \) and corresponding values of the value function that are linearly interpolated. Once the value iteration algorithm converges, the resulting Q-function is approximated by a neural network. The motivation behind this final step is that the number of parameters in a neural network can be significantly less than the number of grid points, and thus memory requirements are reduced for on-line use of the Q-functions.

The continuous state space \([0,1]\) of the belief state \( b_k \) is replaced by a set of grid points \( B = \{b^1, b^2, \ldots, b^{n_b}\} \), where \( 0 = b^1 < b^2 < \ldots < b^{n_b-1} < b^n = 1 \). If the approximate value function \( V^{(i)}(x_k, b_k) \) needs to be evaluated at a non-grid belief state \( b_k \in [b^j, b^{j+1}] \) than the weight \( \xi \in [0,1] \) is found such that \( b^j = \xi b^j + (1 - \xi) b^{j+1} \) and \( V^{(i)}(x_k, b_k) \) is approximated by the linear interpolation

\[
V^{(i)}(x_k, b_k) \approx \xi V^{(i)}(x_k, b^j) + (1 - \xi)V^{(i)}(x_k, b^{j+1}).
\]

(29)

The value obtained by the linear interpolation is always less or equal to the the true value \( V^{(i)}(x_k, b_k) \) because \( V^{(i)}(x_k, b_k) \) is a concave function in the belief state \( b_k \). Although the introduced grid and linear interpolation make the bound (28) inaccurate, it is still a useful rule for terminating the value iteration algorithm [23]. Note that in comparison with the finite time horizon case the number of sample points and function values needed to store approximate value function \( V^{(i)} \) is \((n_x n_u + 1) n_b \) and in the case of the Q-function, it is \((n_x n_u + 1) n_b \). As the required number of the value functions might be prohibitive for on-line use, the final step is to approximate the sampled Q-function by a perceptron neural network [24].

A perceptron neural network with one hidden layer is used for each \( x_k = \bar{x} \) and \( u_k = \bar{u} \), \( \bar{x} \in \mathcal{X}, \bar{u} \in \mathcal{U} \) to approximate the partial Q-function \( Q^{(i)\text{final}}(\bar{x}, b_k, \bar{u}) \) as

\[
Q^{(i)\text{final}}(\bar{x}, b_k, \bar{u}) \approx Q^{a}(\bar{x}, b_k, \bar{u}) = \sum_{j=1}^{N_{NN}} c_{\bar{x}, \bar{u}, j} \tanh(w_{\bar{x}, \bar{u}, j}^T[0,1]),
\]

(30)

where \( N_{NN} \) is the number of neurons in the hidden layer, \( c_{\bar{x}, \bar{u}, j} \in \mathbb{R} \) and \( w_{\bar{x}, \bar{u}, j} \in \mathbb{R}^2 \) are parameters of the neural network. These parameters are determined such that the
least squares criterion

\[ J_{\text{LS}}(v_{x,u,j}, w_{x,u,j}) = \sum_{j=1}^{n_b} (Q^{(i_{\text{final}})}(\bar{x}, b^j, \bar{u}) - Q^*(\bar{x}, b^j, \bar{u}))^2 \]  

(31)

is minimized. Since such a criterion is nonlinear in the parameters, the Levenberg-Marquardt iterative optimization algorithm is used in a batch mode to estimate the parameters.

5 Numerical example

A small-scale numerical example is used to illustrate the proposed approach. A particular attention is paid to the reduction of memory requirements resulting from the use of a neural network as an approximator of the Q-functions.

It is considered that the observed system has the following state space \( \mathcal{X} = \{1, 2, 3\} \) and input space \( \mathcal{U} = \{1, 2\} \). The transition probabilities are as follows

\[
P(x_{k+1} | x_k, u_k = 1) = \begin{bmatrix}
0.9 & 0.8 & 0.05 \\
0.05 & 0.15 & 0.8 \\
0.05 & 0.05 & 0.15 
\end{bmatrix},
\]

\[
P(x_{k+1} | x_k, u_k = 2) = \begin{bmatrix}
0.15 & 0.05 & 0.05 \\
0.8 & 0.15 & 0.05 \\
0.05 & 0.8 & 0.9 
\end{bmatrix},
\]

\[
P(x_{k+1} | x_k, u_k = 1) = \begin{bmatrix}
0.5 & 0.8 & 0.05 \\
0.05 & 0.15 & 0.8 \\
0.45 & 0.05 & 0.15 
\end{bmatrix},
\]

\[
P(x_{k+1} | x_k, u_k = 2) = \begin{bmatrix}
0.15 & 0.05 & 0.42 \\
0.8 & 0.15 & 0.08 \\
0.05 & 0.8 & 0.5 
\end{bmatrix}. \tag{32}
\]

An intermittent fault that might recover itself with the following mode transition probabilities is considered

\[
P(\mu_{k+1} | \mu_k) = \begin{bmatrix}
0.92 & 0.08 \\
0.08 & 0.92 
\end{bmatrix}. \tag{33}
\]

The discount factor \( \lambda = 0.9 \), the required accuracy \( \delta = 1e-3 \) and the number of the grid points \( n_b = 100 \) are used. The approximation of the optimal value function \( V^*(s_k) \) obtained by the value iteration algorithm after \( i_{\text{final}} = 68 \) iterations is depicted in Fig. 4. The corresponding Q-functions and their approximations by neural networks with \( N_{\text{NN}} = 3 \) neurons in the hidden layer are shown in Fig. 5. To store the Q-function obtained from the value iteration algorithm, \( (n_x n_u + 1)n_b = 600 \) values are needed. In comparison, the neural network approximation of the Q-function requires only \( 3n_x n_u N_{\text{NN}} = 54 \) parameters. It represents approximately ten fold decrease in memory requirements and the same quality of the input signal generator is preserved.

6 Conclusion

The paper focused on active fault detection with a given detector over an infinite time horizon. The problem was formulated for a class of systems that can be described using discrete-time finite-state controlled Markov chain with partially observed state and the approximate input signal generator that improves the decision quality of a given fault detector was designed. Besides providing the solution for the infinite time horizon, the resulting input signal generator can be used as an approximation for finite horizon problems where the finite horizon is sufficiently long or uncertain. The main advantage of the proposed approach is a significant reduction of on-line computational demands because the most computation is carried on off-line.

Acknowledgement

This work was supported by the Czech Science Foundation, project No. GAP103/11/P407.

References


Figure 5. Approximate Q-functions $Q^{(68)}(s_k, u_k)$ (solid blue lines) and their neural network representations $Q^a(s_k, u_k)$ (dash-dot red lines).


