# Temporal-Difference Q-learning in Active Fault Diagnosis

Jan Škach [1] Ivo Punčochář [1] Frank L. Lewis [2]

[1] Identification and Decision Making Research Group (IDM)
European Centre of Excellence - NTIS
University of West Bohemia
Pilsen, Czech Republic

[2] Advanced Controls and Sensors Group (ACS)
UTA Research Institute UTARI
University of Texas at Arlington
Ft. Worth, TX, USA


3rd International Conference on Control and Fault-Tolerant Systems,
Barcelona, Spain

# Outline

# Introduction

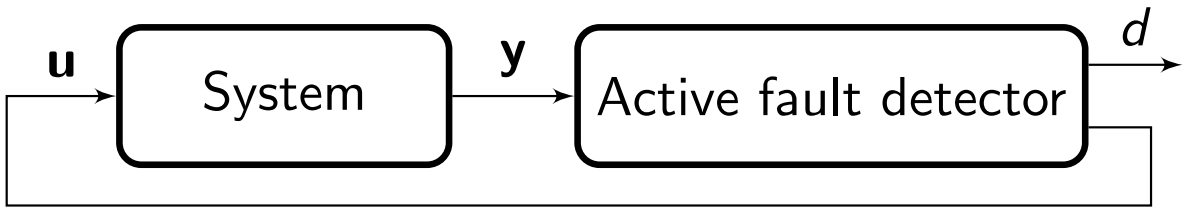
## Fault detection

- Fault detection helps to improve system reliability or reduce operating costs.
- Important methods of fault detection are model-based methods.
- One group of model-based methods is based on multiple-model fault detection. The aims is to decide correctly about a model from a set of all models that describe a possible behavior of a system.

## Passive fault detection

- The decision about model is generated based on the input and output data. There is no action of a passive detector on a system.
- Since the passive fault detection architecture does not influence a system, some faults become evident after unacceptably long time-period.

# Introduction




## Active fault detection

- Active fault detection (AFD) is based on an idea of improving the quality of detection by probing the monitored system by a suitably designed input signal **u**.
- AFD methods can be classified into two groups.
  - Deterministic methods consider system disturbances to be bounded signals.
  - Probabilistic methods assume disturbances to be random variables with known probabilistic distributions.
- A probabilistic AFD method based on minimization of a general detection cost criterion over an infinite-time horizon is discussed.

# Problem formulation

*Imperfect state information problem*

Model of system

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}_{\mu_k}\mathbf{x}_k + \mathbf{B}_{\mu_k}\mathbf{u}_k + \mathbf{G}_{\mu_k}\mathbf{w}_k,\\ \mathbf{y}_k &= \mathbf{C}_{\mu_k}\mathbf{x}_k + \mathbf{H}_{\mu_k}\mathbf{v}_k,\\ \mathbf{P}_{j,i} &= P(\mu_{k+1}=j \mid \mu_k = i)\end{aligned}$$

Estimator of $\mathbf{x}_k$ and $\mu_k$

Criterion

$$J = \lim_{F\to\infty} \mathrm{E}\left\{\sum_{k=0}^{F} \eta^k L^{\mathrm{d}}(\mu_k, d_k)\right\}$$

Active fault detector

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \boldsymbol{\rho}(\mathbf{y}_0^k, \mathbf{u}_0^{k-1})$$

State $\mathbf{x}_k$ and model index $\mu_k$ are not directly accessible.

**Problem reformulation**

*Perfect state information problem*

Model of system and estimator

$$\boldsymbol{\xi}_{k+1} = \phi(\boldsymbol{\xi}_k, \mathbf{u}_k, \mathbf{y}_{k+1})$$

Criterion

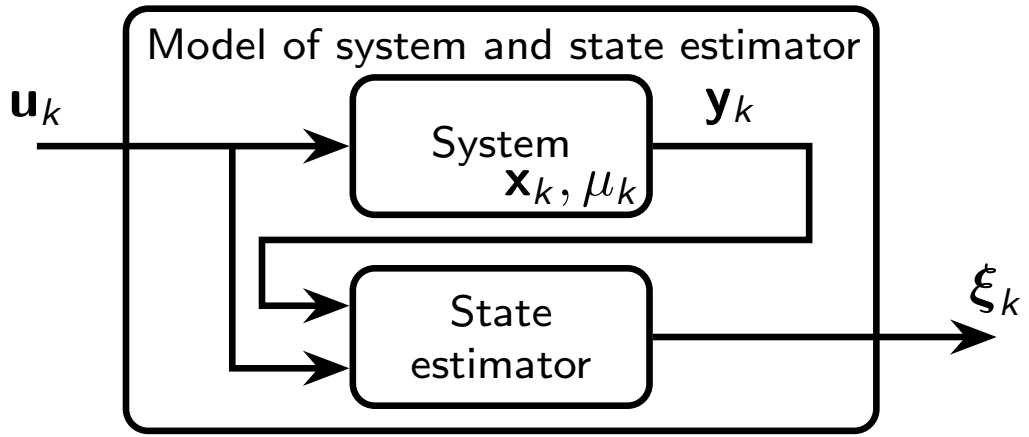
$$J = \lim_{F\to\infty} \mathrm{E}\left\{\sum_{k=0}^{F} \eta^k \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}_k, d_k)\right\}$$

Active fault detector

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\boldsymbol{\rho}}(\boldsymbol{\xi}_k)$$

Hyper state $\boldsymbol{\xi}_k$ represents sufficient statistics of $\mathbf{x}_k$ and $\mu_k$.

# Problem formulation




## Model of system and state estimator

$$\xi_{k+1} = \phi(\xi_k, \mathbf{u}_k, \mathbf{y}_{k+1}),$$

where $k \in \mathcal{T} = \{0, 1, \ldots\}$ is a time index, $\xi_k \in \mathcal{G} \subset \mathbb{R}^{n_\xi}$ is a hyper state, $\mathbf{u}_k \in \mathcal{U} = \{\bar{\mathbf{u}}^1, \ldots, \bar{\mathbf{u}}^{N_u}\} \subset \mathbb{R}^{n_u}$ is an input, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is an output, and $\phi : \mathcal{G} \times \mathcal{U} \times \mathbb{R}^{n_y} \mapsto \mathcal{G}$ is a function that describes a behavior of the original system coupled with a state estimator.

- The state estimator consists of a bank of Kalman filters and Generalized Pseudo Bayes algorithm that tracks only a $h$-step history of possible model sequences and reduce computational complexity.

# Problem formulation

## Active fault detector

The goal is to find an active fault detector represented by a stationary policy $\bar{\boldsymbol{\rho}} : \mathcal{G} \mapsto \mathcal{M} \times \mathcal{U}$ that generates an input $\mathbf{u}_k$ and a decision $d_k \in \mathcal{M} = \{1, 2, \ldots, N_\mu\}$ about the model index $\mu_k \in \mathcal{M}$,

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\boldsymbol{\rho}}(\boldsymbol{\xi}_k) = \begin{bmatrix} \bar{\sigma}(\boldsymbol{\xi}_k) \\ \bar{\boldsymbol{\gamma}}(\boldsymbol{\xi}_k) \end{bmatrix},$$

where $\bar{\sigma} : \mathcal{G} \mapsto \mathcal{M}$ is a stationary policy of the decision generator and $\bar{\boldsymbol{\gamma}} : \mathcal{G} \mapsto \mathcal{U}$ is a stationary policy of the input signal generator.

## Criterion

$$J = \lim_{F \to \infty} \mathrm{E} \left\{ \sum_{k=0}^{F} \eta^k \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}_k, d_k) \right\},$$

where $\bar{L}^{\mathrm{d}} : \mathcal{G} \times \mathcal{M} \mapsto \mathbb{R}^+$ is a given detection cost function that imposes a penalty on incorrect decisions, $\eta \in (0, 1)$ is a discount factor, and $\mathrm{E}\{\cdot\}$ is the expectation operator.

# Dynamic programming solution

## Dynamic programming

The optimal value function $V^* : \mathcal{G} \mapsto \mathbb{R}$ satisfies the Bellman functional equation

$$V^*(\boldsymbol{\xi}) = \min_{d \in \mathcal{M}, \mathbf{u} \in \mathcal{U}} \mathrm{E}\left\{ \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}, d) + \eta V^*(\phi(\boldsymbol{\xi}, \mathbf{u}, \mathbf{y}')) | \boldsymbol{\xi}, \mathbf{u}, d \right\}.$$

- The Bellman functional equation is almost impossible to solve analytically. Thus, numerical algorithms are used such as the policy iteration algorithm.
- Due to a size of the hyper-state space $\mathcal{G}$, suboptimal techniques such as a state-space quantization or linear value function approximation (VFA) must be employed.
- It is not easy to find the VFA by dynamic programming. However, reinforcement learning could naturally identify the most important regions of the hyper-state space for which the VFA could be subsequently obtained.

# Reinforcement learning solution

## Q-function

In reinforcement learning, a Q-function is used. The optimal Q-function $Q^* : \mathcal{G} \times \mathcal{U} \times \mathcal{M} \mapsto \mathbb{R}$ for the AFD problem can be defined as

$$Q^*(\boldsymbol{\xi}, \mathbf{u}, d) = \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}, d) + \eta \, \mathrm{E}\left\{ V^*(\boldsymbol{\phi}(\boldsymbol{\xi}, \mathbf{u}, \mathbf{y}')) | \boldsymbol{\xi}, \mathbf{u} \right\}.$$

## Q-function approximation

An approximation $\tilde{Q} : \mathcal{G} \times \mathcal{U} \times \mathcal{M} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ of the Q-function can be defined as

$$\tilde{Q}(\boldsymbol{\xi}, \mathbf{u}, d, \boldsymbol{\theta}) = \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}, d) + \sum_{i=1}^{n_\theta} \psi_i(\boldsymbol{\xi}, \mathbf{u}) \theta_i = \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}, d) + \boldsymbol{\psi}(\boldsymbol{\xi}, \mathbf{u})^{\mathsf{T}} \boldsymbol{\theta},$$

where $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{n_\theta}]^{\mathsf{T}} \in \mathbb{R}^{n_\theta}$ is a vector of weights and
$\boldsymbol{\psi} : \mathcal{G} \times \mathcal{U} \mapsto \mathbb{R}^{n_\theta}$ is a vector-valued function of basis functions.

# Reinforcement learning solution

## Temporal-difference Q-learning

- A temporal difference (TD) Q-learning is a method of learning the active fault detector iteratively from experience.
- The decision $d_k$ and the auxiliary input $\mathbf{u}_k$ are generated by the approximate policy $\tilde{\boldsymbol{\rho}}^{(m)} = [\bar{\sigma}^*,\ (\tilde{\boldsymbol{\gamma}}^{(m)})^{\mathsf{T}}]^{\mathsf{T}}$ according to

$$d_k = \bar{\sigma}^*(\boldsymbol{\xi}_k) = \arg\min_{d \in \mathcal{M}} \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}_k, d),$$

$$\mathbf{u}_k = \tilde{\boldsymbol{\gamma}}^{(m)}(\boldsymbol{\xi}_k) = \arg\min_{\mathbf{u} \in \mathcal{U}} \left\{ \tilde{Q}\left(\boldsymbol{\xi}_k, \mathbf{u}, \bar{\sigma}^*(\boldsymbol{\xi}_k), \boldsymbol{\theta}^{(m)}\right) \right\},$$

  where $m = 0, 1, \ldots,$ is an iteration index.
- The weights $\boldsymbol{\theta}^{(m)}$ of the approximate Q-function $\tilde{Q}\left(\boldsymbol{\xi}_k, \mathbf{u}, \bar{\sigma}^*(\boldsymbol{\xi}_k), \boldsymbol{\theta}^{(m)}\right)$ are updated using a TD error $\delta_k^Q$.

# Reinforcement learning solution

## Temporal-difference Q-learning

- The TD-error expresses a difference between the expected costs and current admitted costs based on the simulation data,

$$\delta_k^Q = \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}_k, d_k) + \eta \tilde{Q}\left(\boldsymbol{\xi}_{k+1}, \mathbf{u}_{k+1}, \bar{\sigma}^*(\boldsymbol{\xi}_{k+1}), \boldsymbol{\theta}^{(m)}\right) - \tilde{Q}\left(\boldsymbol{\xi}_k, \mathbf{u}_k, \bar{\sigma}^*(\boldsymbol{\xi}_k), \boldsymbol{\theta}^{(m)}\right).$$

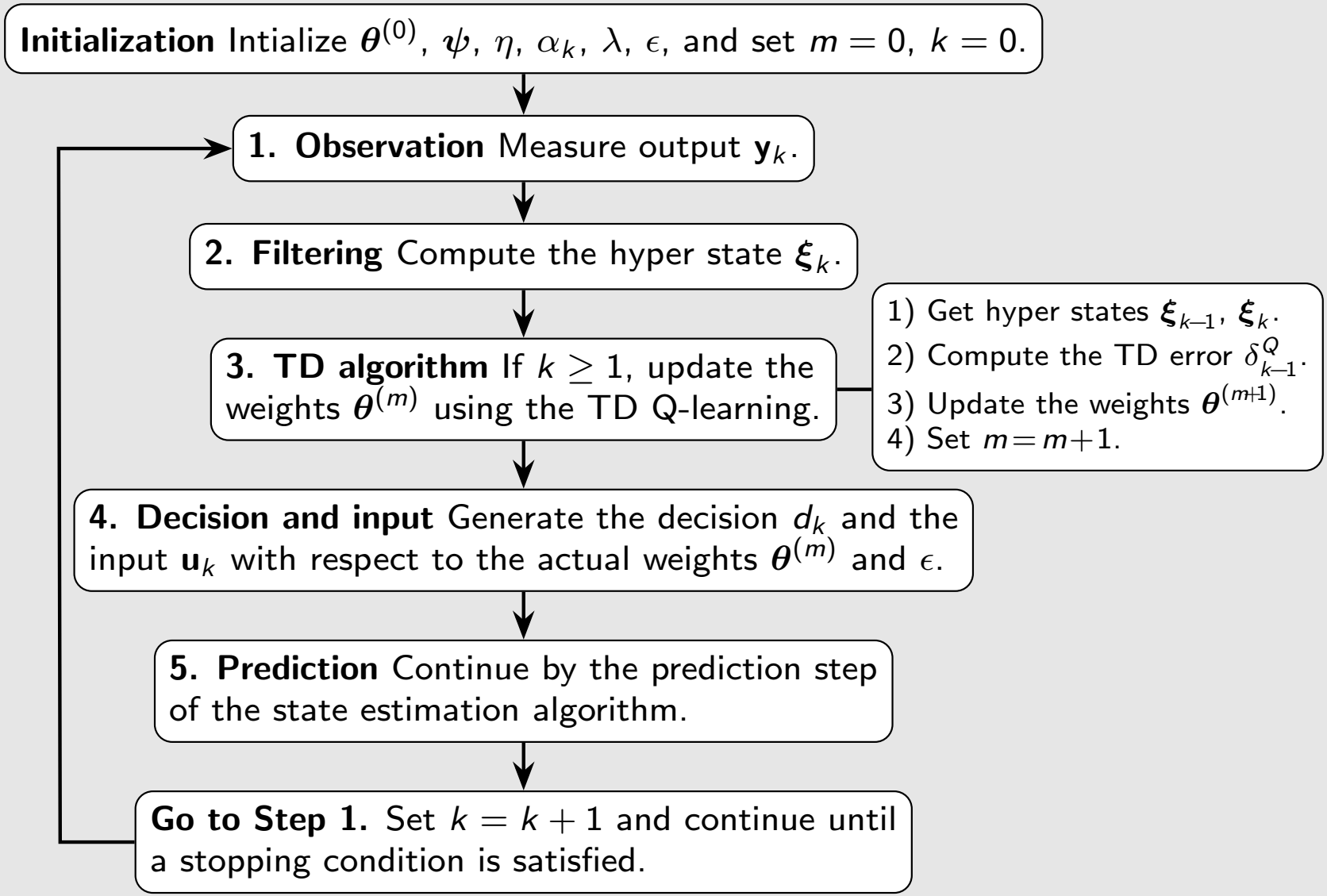
- The weights $\boldsymbol{\theta}^{(m)}$ can be updated as

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + \alpha_k \delta_k^Q \mathbf{z}_k^Q,$$

  where $\alpha_k > 0$ is a scalar step-size parameter, $\lambda \in [0, 1]$ is a TD parameter, and $\mathbf{z}_k^Q \in \mathbb{R}^{n_\theta}$ is an eligibility vector recursively defined as $\mathbf{z}_{k+1}^Q = \eta\lambda \mathbf{z}_k^Q + \boldsymbol{\psi}(\boldsymbol{\xi}_{k+1}, \mathbf{u}_{k+1})$.
- Exploration of the hyper-state space can be supported by assuming an $\epsilon$-greedy policy, $0 \leq \epsilon \leq 1$.

# Reinforcement learning solution

## Temporal-difference Q-learning algorithm



# Numerical example

## A second-order linearized model of a pendulum

$$\begin{bmatrix} \mathbf{x}_{1,k+1} \\ \mathbf{x}_{2,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 1 - \frac{T_s g}{l} & \frac{-T_s \beta_{\mu_k}}{ml^2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1,k} \\ \mathbf{x}_{2,k} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T_s}{ml^2} \end{bmatrix} u_k + \mathbf{G}\mathbf{w}_k,$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} + \mathbf{H}\mathbf{v}_k,$$

where $\mathbf{x}_{1,k}$ [rad] is an angle of displacement from the zero downward position, $\mathbf{x}_{2,k}$ [rad$\cdot$s$^{-1}$] is an angular velocity, $u_k \in \{-10, 0, 10\}$ [N$\cdot$m] is a moment of a force applied at the pendulum joint, $\beta_1 = 6$ [kg $\cdot$ m$^2$ $\cdot$ s$^{-1}$] is a friction coefficient, $l = 1$ [m] is a length of pole, $m = 2$ [kg] is a mass of pendulum, $g = 9.81$ [m$\cdot$s$^{-2}$] is the gravitational acceleration, $T_s = 5 \cdot 10^{-2}$ [s] is a sampling period, $\mathbf{G} = 8 \cdot 10^{-4}\mathbf{I}_2$, $\mathbf{H} = 10^{-3}$, $P(\mu_{k+1} = j | \mu_k = i) = 0.02$ for $i, j \in \{1, 2\}$, $i \neq j$ are transition probabilities, and the initial conditions are $\hat{\mathbf{x}}_{0|-1}^{\mathsf{T}} = [0\ 0]$, $\mathbf{\Sigma}_{0|-1}^{\mathbf{x}} = 2 \cdot 10^{-4}\mathbf{I}_2$, and $P(\mu_0 = 1) = 1$.
In case of the faulty behavior the friction coefficient changes to $\beta_2 = 6.2$ [kg $\cdot$ m$^2$ $\cdot$ s$^{-1}$].

# Numerical example

## Simulation example settings

The detection cost function is defined as

$$L^{\mathrm{d}}(\mu_k, d_k) = \begin{cases} 0 & \text{if } d_k = \mu_k, \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

- The discount factor is $\eta = 0.98$ and the $h$-step history is $h = 1$.
- A performance is studied for a zero constant signal (ZSG), sine signal with amplitude 10 and frequency 1 [Hz] (SSG), active fault detector designed by a TD learning (AFDR) that uses 21 normalized Gaussian basis functions to approximate the value function and 100 Monte Carlo (MC) simulations to approximate the expectation, and active fault detector designed by TD Q-learning (AFDRQ) that uses 63 normalized Gaussian basis functions, and the exploration parameter $\epsilon = 0.03$. Both detectors are tuned in 10000 time steps with parameters $\alpha_k = \frac{1000}{2000+k}$ and $\lambda = 0.4$.

# Numerical example

## Simulation results

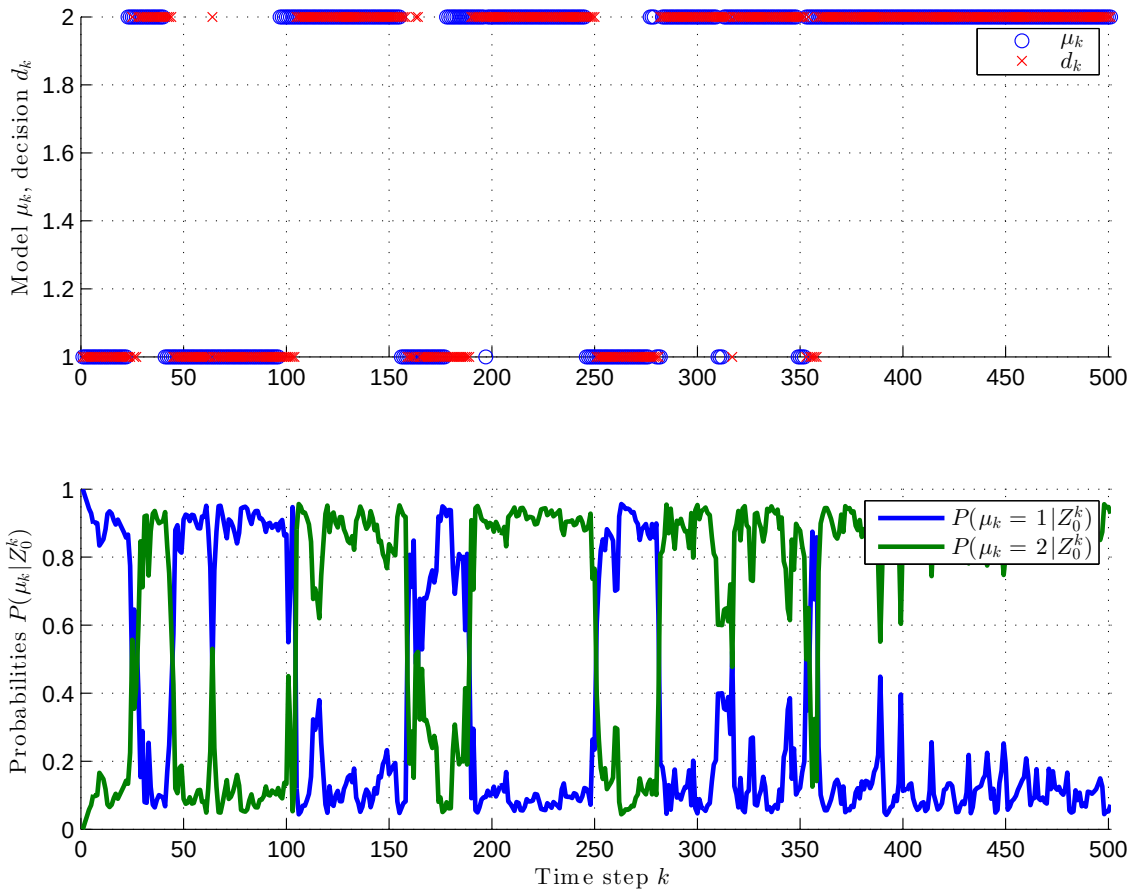
- The performance, in terms of estimates of the criterion $\hat{J}$ and variance $\text{var}\{\hat{J}\}$, of the designed active fault detector compared to the other input signals is evaluated through 1000 MC simulations.

| Input signal generator | $\hat{J}$ | $\text{var}\{\hat{J}\}$ |
|---|---|---|
| ZSG | 1.3119 | 0.0043 |
| SSG | 1.0942 | 0.0029 |
| AFDR | 1.0239 | 0.0025 |
| AFDRQ | 0.9879 | 0.0023 |

# Numerical example

## Simulation results

- Typical trajectories of the model $\mu_k$, decision $d_k$, and conditional model probabilities $P(\mu_k|\mathbf{I}_0^k)$ for the AFDRQ.



# Conclusion

## Contributions and conclusion

- A problem of active fault detection for stochastic linear Markovian switching systems on the infinite-time horizon is considered.
- An active fault detector that minimizes a general detection cost criterion is designed.
- A simulation-based algorithm based on the TD Q-learning is proposed and its good performance is shown in the numerical example.
- The future step is to extend the problem to nonlinear systems and apply the proposed algorithm to find the active fault detector.
- Another direction in the future can be better analysis of the basis function selection and convergence.

# Thank you!

contact: http://idm.kky.zcu.cz

# References

A list of references relevant to the topic.


1. Punčochář, I., Škach, J., and Šimandl, M. (2015). Infinite Time Horizon Active Fault Diagnosis Based on Approximate Dynamic Programming. In *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, 4456-4461. Osaka, Japan.