Optimal Active Fault Diagnosis by Temporal-Difference Learning

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

55th IEEE Conference on Decision and Control, Las Vegas, USA

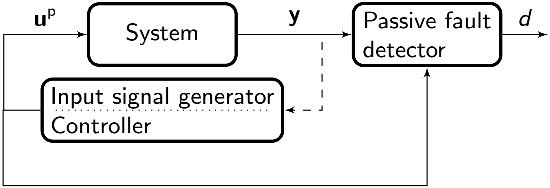
Outline

Fault detection

- Fault detection improves system reliability or reduces operating costs.
- Important methods of fault detection are model-based methods.
- One group of model-based fault detection methods is based on a multiple-model approach. The aim is to decide correctly about a model from a set of all models that represent fault-free and faulty behavior of a system.
- Fault detection methods can be classified as
  - passive
  - active.

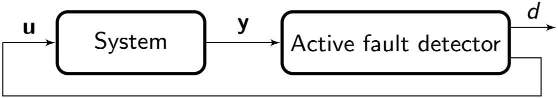
Passive fault detection

- 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 could become evident after unacceptably long time-period.



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**.
- 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 μ_k

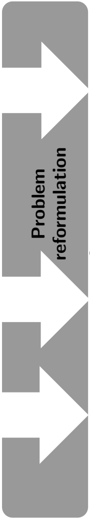
Criterion

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

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 μ_k are not directly accessible.



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 $\mathbf{y}_0^k$ and $\mathbf{u}_0^{k-1}$.

Problem formulation

Perfect state information problem

The **goal** of the perfect state information optimization problem is to find a stationary policy $\bar{\boldsymbol{\rho}}$ that generates the input $\mathbf{u}_k$ and decision d_k,

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

The **solution** can be found using dynamic programming.

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 $\mathbf{y}_0^k$ and $\mathbf{u}_0^{k-1}$.

Dynamic programming solution

Dynamic programming

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

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

- 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.
- Reinforcement learning could naturally identify the most important regions of the hyper-state space for which the VFA could be subsequently obtained.

Value function approximation

A linear parametric approximation $\tilde{V} : \mathcal{G} \times \mathbb{R}^{n_\omega} \mapsto \mathbb{R}$ of the value function can be defined as

$$\tilde{V}(\boldsymbol{\xi}, \boldsymbol{\omega}) = \sum_{i=1}^{n_\omega} \varphi_i(\boldsymbol{\xi})\omega_i = \boldsymbol{\varphi}(\boldsymbol{\xi})^\mathsf{T}\boldsymbol{\omega},$$

where $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_{n_\omega}]^\mathsf{T} \in \mathbb{R}^{n_\omega}$ is a vector of weights and $\boldsymbol{\varphi} : \mathcal{G} \mapsto \mathbb{R}^{n_\omega}$ is a vector-valued function of basis functions $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_{n_\omega}]^\mathsf{T}$.

- The weights $\boldsymbol{\omega}$ must be found. This is done by temporal-difference learning.

Reinforcement learning solution

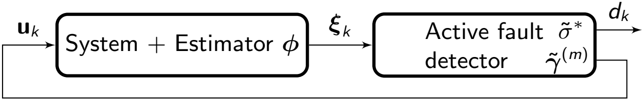
Temporal-difference learning

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

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

$$\mathbf{u}_k = \tilde{\gamma}^{(m)}(\boldsymbol{\xi}_k) = \arg\min_{\mathbf{u} \in \mathcal{U}} \mathsf{E}\left\{\tilde{V}\left(\phi(\boldsymbol{\xi}_k, \mathbf{u}, \mathbf{y}'), \boldsymbol{\omega}^{(m)}\right) |\boldsymbol{\xi}_k, \mathbf{u}\right\},$$

  where $m = 0, 1, \ldots,$ is an iteration index.
- The weights $\boldsymbol{\omega}^{(m)}$ of the approximate value function $\tilde{V}$ are updated using a TD error δ_k.



Temporal-difference learning

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

$$\delta_k = \bar{L}^{\mathrm{d}}(\boldsymbol{\xi}_k, d_k) + \eta \tilde{V}(\boldsymbol{\xi}_{k+1}, \boldsymbol{\omega}^{(m)}) - \tilde{V}(\boldsymbol{\xi}_k, \boldsymbol{\omega}^{(m)}).$$

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

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

  where $\alpha_k > 0$ is a scalar step-size parameter, $\lambda \in [0, 1]$ is a TD parameter, and $\mathbf{z}_k \in \mathbb{R}^{n_\omega}$ is an eligibility vector recursively defined as $\mathbf{z}_{k+1} = \eta\lambda\mathbf{z}_k + \boldsymbol{\varphi}(\boldsymbol{\xi}_{k+1})$.
- Exploration of the hyper-state space can be supported by adding a deterministic exploration signal $\boldsymbol{\vartheta} \in \mathcal{T} \mapsto \mathbb{R}^{n_u}$ to auxiliary input signal, i.e., $\mathbf{u}_k = \tilde{\gamma}^{(m)}(\boldsymbol{\xi}_k) + \boldsymbol{\vartheta}_k$.

Temporal-difference learning algorithm

Initialization Intialize $\boldsymbol{\omega}^{(0)}$, $\boldsymbol{\varphi}$, η, α_k, λ, $\boldsymbol{\vartheta}$, and set $m = 0$, $k = 0$.

1. Observation Measure output $\mathbf{y}_k$.

2. Filtering Compute the hyper state $\boldsymbol{\xi}_k$.

3. TD algorithm If $k \geq 1$, update the weights $\boldsymbol{\omega}^{(m)}$ using the TD learning.

1) Get hyper states $\boldsymbol{\xi}_{k-1}$, $\boldsymbol{\xi}_k$.
2) Compute the TD error δ_{k-1}.
3) Update the weights $\boldsymbol{\omega}^{(m+1)}$.
4) Set $m = m+1$.

4. Decision and input Generate the decision d_k and the input $\mathbf{u}_k$ with respect to the actual weights $\boldsymbol{\omega}^{(m)}$ and $\boldsymbol{\vartheta}$.

5. Prediction Continue by the prediction step of the state estimation algorithm.

Go to Step 1. Set $k = k + 1$ and continue until a stopping condition is satisfied.

A second-order linearized model of a pendulum

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

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

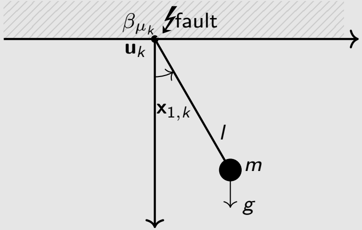
where $\mathbf{x}_{1,k}$ [rad] is an angle of displacement from the zero downward position,
$\mathbf{x}_{2,k}$ [rad$\cdot$s^{-1}] is an angular velocity,
$u_k \in \{-10, 0, 10\}$ [N$\cdot$m] is a moment of a force applied at the pendulum joint,
$\beta_1 = 6$ [kg $\cdot$ m^2 $\cdot$ s^{-1}] is a friction 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,
$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_x})$ and $\mathbf{v}_k \sim \mathcal{N}(0,1)$ are mutually independent noises, $\mathbf{G} = 8\cdot 10^{-4}\mathbf{I}_2$, $\mathbf{H} = 10^{-3}$,
$P(\mu_{k+1} = j | \mu_k = i) = 0.02$ for $i, j \in \{1, 2\}$, $i \neq j$ are transition probabilities.
The initial conditions are $\hat{\mathbf{x}}^{\mathrm{T}}_{0|-1} = [0\ 0]$, $\mathbf{\Sigma}^{\mathbf{x}}_{0|-1} = 2\cdot 10^{-4}\mathbf{I}_2$, and $P(\mu_0 = 1) = 1$.

- In case of the faulty behavior the friction coefficient changes to $\beta_2 = 6.2$.
- State estimation is performed by a bank of Kalman filters with the Generalized Pseudo-Bayes algorithm tracking the history $h = 1$ of possible model sequences.

A second-order linearized model of a pendulum

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

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



- In case of the faulty behavior the friction coefficient changes to $\beta_2 = 6.2$.
- State estimation is performed by a bank of Kalman filters with the Generalized Pseudo-Bayes algorithm tracking the history $h = 1$ of possible model sequences.

Simulation example settings

The detection cost function is defined as

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

- The discount factor is $\eta = 0.98$.
- The value function is approximated by 25 normalized Gaussian basis functions, the initial weights are $\boldsymbol{\omega}^{(0)} = \mathbf{0}$, and 100 Monte Carlo (MC) simulations to approximate the expectation of $\tilde{V}$.
- The active fault detector is tuned in 10000 time steps with parameters $\alpha_k = \frac{1000}{2000+k}$ and $\lambda = 0.4$.
- A performance is studied and compared to a zero constant signal (ZISG) and sine signal with amplitude 10 and frequency 1 [Hz] (SISG).

Simulation results

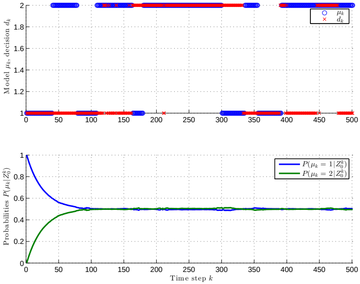
- The performance, in terms of estimates of the criterion $\hat{J}$ and variance var$\{\hat{J}\}$ on the finite horizon of 501 time steps, of the designed active fault detector compared to the other input signals is evaluated through 1000 MC simulations.

Input signal generator	$\hat{J}$	var$\{\hat{J}\}$
ZISG	1.4156	0.0048
SISG	1.1734	0.0033
AFDR	1.1219	0.0028

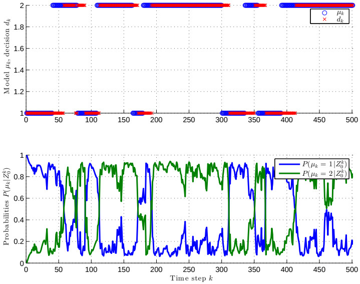
AFDR has the lowest values of $\hat{J}$ and var$\{\hat{J}\}$.

Simulation results

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



ZISG (passive detection)



AFDR (active detection)

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 temporal difference learning is proposed and its good performance is shown in the numerical example.
- One direction of the future research can aim at analyzing the basis function selection and convergence properties.

Thank you!

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