Fault Tolerant Control Using Reinforcement Learning and Particle Swarm Optimization

DAPENG ZHANG,1 AND ZHIWEI GAO2 (Senior Member, IEEE)

1School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
2Faculty of Engineering and Environment, University of Northumbria, Newcastle upon Tyne NE2 8ST, U.K.

Corresponding author: Dapeng Zhang (zdp@tju.edu.cn)

This work was supported in part by the Tianjin University Innovation Fund under Grant 2020XT-0024, in part by the Chinese Scholarship Council (CSC) Overseas Visiting Project, and in part by the Alexander von Humboldt Foundation under Grant GRO/1117303 STP.

ABSTRACT Diversity, uncertainty and suddenness of unexpected faults bring a challenge for fault tolerant control due to the lack of valid data especially for a fault during an early stage. In this study, a reinforcement learning approach with a critic action architecture is proposed to overcome this challenge by designing an online learning fault-tolerant controller so that the faulty system can approximate the performance index of the fault-free system. Different from the traditional Hebb enhancement rules in the reinforcement learning, the training process is speeded up by introducing a supervisory learning on the basis of the training dataset which is built with the states and the virtual optimal control acquired by particle swarm optimization. The effectiveness of the algorithm is demonstrated by a test bed of a three-tank system.

INDEX TERMS Fault tolerant control, reinforcement learning, particle swarm optimization.

I. INTRODUCTION

Fault tolerant control (FTC) has received much attention in engineering applications due to its prominent guidance of system reliability and safety, high system performance, productivity and operating efficiency [1]. Great efforts have been made to mitigate the influences from the faults on the operating system based on hard-ware and/or information redundancy based tolerant control strategies [2]–[7]. Generally, the FTC approaches are classified into passive FTC and active FTC [8], and model-based FTC played a dominant role during the past decades. Recently, fuzzy adaptive finite-time fault-tolerant control [9] and event-triggered robust fuzzy adaptive finite-time control [10] approaches were investigated respectively. It is noted that model-based FTC highly depends on the accuracy of the model as an excellent model is composed of accurate and sufficient information on a research object such as parameters and structures, which enables an effective control. However, the model would be unavailable or difficult to build for a complex industrial process. Encouragingly, data-based methods [11]–[14] provide a feasible way to complement the model-based approaches based on abundant data that directly reflect the actual status of system. Among them the machine learning and artificial intelligence such as neural network (NN) [15], support vector machine (SVM) [16] and deep learning [17] are striking by imitating the human brain and other intelligent agents. These intelligent approaches use “learning” to grasp the inner rules based on the training data. Usually there are supervised methods and unsupervised methods for “learning” in data-driven fault diagnosis [18]. The supervised learning has a higher efficiency of learning based on a training set that requires a number of data to cover the relations from inputs to goals. The unsupervised learning is essentially a data clustering based on certain criterion (usually norm) that also needs enough data to guard the pattern’s accuracy though it does not need a prior training set.

In principle it is impossible to get all kinds of fault data due to the constraint of physical condition and the economic consideration. On the other hand the data of FTC are imbalance in which plenty of data are healthy and a few are faulty. How to find a way to learn by using data as little as possible is well motivated for FTC. The reinforcement learning (RL) brings a novel way to learn by introducing an idea of control value instead of control effect. The control value is evaluated immediately after each action. As a result, the “learning” becomes a process of seeking the best evaluation of control which does not require all kinds of faulty data in advances. It only needs the current state and the next state at every trial. After a trial and error that simulates the process of animals’
adapting to environment [19], [20], the controller obtains an
ability of keeping consistency of the performance index and
the working condition by action’s self-adjusting.

The RL provides a feasible approach to learn in unknown
environment without need of abundant prior data. It is ben-
eficial to the FTC of lacking early faulty data. As a result,
if one takes the fault as environment, the healthy performance
index as a goal, the FTC control rule will be achieved by
self-adjusting with RL approach. Recently, RL based FTC
have paid attention and some results were reported such as
FTC tracking control for MIMO discrete-time systems [21],
FTC design for a class of nonlinear MIMO discrete-time sys-
tems [22] and RL-based FTC for linear discrete-time dynamic
systems [23]. It is noticed that learning of RL is a very slow
process which makes RL be limited to a small FTC
system [24] although RL has succeeded in game, computer
vision [25] and robot [26]. Several methods accelerating on
RL which aim to a huge system have proposed such as traject-
ory trace [20], trust region policy optimization(TRPO) [27],
actor-critic with experience replay (ACER) [28], proximal
policy optimization algorithm(PPO) [29], FPGA [30] and so
forth. However, all these methods need a large of training
data, which backs to the dilemma of lack valid training data in
FTC. It is necessary for RL to look for a training accelerating
way to meet the requirements for FTC to mitigate the faulty
influence as quickly as possible.

Compared with the origination from imitating the primary
behavior of animals that looks for a better action by trials
and errors with a low efficiency, the biological actions have
evolved some effective searching methods such as a social
behavior after they have evolved for many millenniums in the
racing for surviving with environment. The particle swarm
optimization (PSO) [31]–[33] which imitating the social
behavior of the bird flock provides a way to find the goal
without needing more data. Therefore, it would be interesting
to select the PSO as an auxiliary tool to improve RL’s learning
speed owing to its excellent parallel random searching ability
in unknown surrounding.

Motivated by the above ideas a reinforcement learning
fault tolerant control via PSO is proposed. A critic action
architecture [34]–[36] which is an efficient way of RL is
proposed to implement FTC by a combination of a critic
network which provides an evaluation of the control value
to meet the goal and an action network which provides the
adjustment of control action. The learning of critic action
is accelerated by converting a training process of the critic
network and the action network alternatively with Hebb
rule to an efficient supervised training via an improved
PSO. The advantages of proposed approach are given as
following:

1) This approach overcomes the conflict between the
required data and no access to get data for all kinds of
faulty data in advance.

2) The optimal control for the best reachable performance
index will be obtained by employing the critic action
regardless of system model and fault style.

3) The responding time for fault is shorten by introduc-
ing the supervisory learning to train the critic action
with PSO.

II. PROBLEM DESCRIPTION AND PRIMARIES

A. PROBLEM DESCRIPTION

Suppose a fault-free nonlinear system/process is described by
\[ x_p(k + 1) = f(x_p(k), u_p(k)) \] (1)
where \( x_p \) is a state of system, \( u_p \) is a control vector.

The optimal control for the best reachable performance
index \( J \) is defined as a summary of immediate cost
\[ J = \sum_{k=0}^{\infty} \gamma^k \sum_{i} R(x_i) \] (6)
where \( \gamma \) is a discount factor with \( 0 \leq \gamma < 1 \) in order to be
convergent.

One will get the optimal controls by carrying an alternation
of the policy evaluation and policy improvement with the formulas (10) and (11)
\[ V_k(x(k)) = R_k(x(k + 1), x(k), u(k)) + \gamma V_k(x(k + 1)) \] (7)

where \( R_k \) is the immediate cost, \( \gamma \) is the discount factor.
\[ u_k(x, u) = \arg\min_u R_k(x(k + 1), x(k), u(k)) + \gamma V_k(x(k + 1)) \] (8)

Formulas (7) and (8) provide a feasible way to get the optimization by only using the state information and the immediate cost. In fact it is not solved directly because \(V_k\) and \(V_{k+1}\) need the information of \(x\) after \(k + 1\) according to (6). Therefore, critic action architecture is proposed to implement it. In critic action architecture a critic network is used to approach to the control value \(V_k\) and \(V_{k+1}\) with input \(x(k)\) and \(x(k+1)\) under the control \(u\). The critic is trained forward in time by trying to minimize the error measure \(\sum_{k=1}^{\infty} \text{TE}^2(k)\) over time. Here

\[ \text{TE}(k) = V_k(X(k)) - R_k(x(k + 1), x(k), u(k)) - \gamma V_k(X(k+1)) \] (9)

where \(X(k)\) stands for either a state vector \(x(k)\) or a concatenation of \(x(k)\) and a control vector \(u(k)\). An action network is used to build a map from input \(x(k)\) to output \(u(k)\). One will get the partial derivative of \(V_k(X(k))\) with respect to the control variable by back propagating through the critic network and then use the gradient of the cost function with respect to the action’s weights to gain the parameters variable of action network for all inputs. The critic action architecture is proposed by Werbos [34] and expressed as heuristic dynamic programming (HDP) and action dependent heuristic dynamic programming (ADHD) in [35]. More details are omitted due to the limitation of space.

**C. CONVERGENCE OF HDP**

The convergence of HDP is given in [37]. Here, it is repeated only for integrity and the proof is omitted.

**Lemma 1:** Consider sequences \(V_k\) and \(u_k\) defined by (10) and (11)

\[ u_k(x(k)) = \arg\min_u \left( x(k)^T M x(k) + u(k)^T N u(k) + V_k(x(k + 1)) \right) \] (10)

\[ V_{k+1}(x(k)) = \min_u \left( x(k)^T M x(k) + u(k)^T N u(k) + V_k(x(k + 1)) \right) \] (11)

If \(V_0(x(k)) = 0\), then it follows that \(V_k\) is a non-decreasing sequence \(V_k : V_{k+1}(x(k)) \geq V_k(x(k))\). Moreover, as \(k \to \infty\), \(V_k \to V^*\), \(u_k \to u^*\), and hence, the sequence \(V_k\) converges to the solution of the discrete-time Hamilton–Jacobi–Bellman.

**Remark 1:** The Hamilton–Jacobi–Bellman is the goal the convergence of the HDP though it cannot be solved exactly in a general nonlinear case.

**III. THE CRITIC ACTION-BASED FTC**

The goal of FTC is to approximate the performance index under fault free condition, by seeking a control \(u\) in the case of unexpected fault. A critic action-based fault-tolerant control (CAFTC) which suits for all kinds of faults without any prior faulty information is proposed and the schematic block-diagram is depicted in figure 1.

**FIGURE 1. Schematic block-diagram of the CAFTC.**

The CAFTC is composed of three parts: a critic network, an action network and a healthy model. In Figure 1 the same critic network is drawn to two boxes in order to express the different samples of a time series. There is no requirement for neural network type to the critic action architecture. As a result, the forward neural network is selected as a critic network and an action network because it is the base of other neural networks.

For a critic network, denoted \(W_{c1}\) as the connected weight between the input layer and hidden layer, \(W_{c2}\) as the connected weight between the hidden layer and output layer, the sigmoid function as the activation function with form of formula (12)

\[ s_{out} = \text{sig}(s_{in}) = \frac{1}{1 + e^{-s_{in}}} \] (12)

whose derivative form is

\[ \dot{s}_{out} = s_{out}(1 - s_{out}) \] (13)

where \(s_{in}\) and \(s_{out}\) are the input and output of the sigmoid function.

As a result, the estimation \(\hat{V}_k(x(k))\) of \(V_k(x(k))\) and \(\hat{V}_{k+1}(x(k+1))\) of \(V_k(x(k + 1))\) are obtained according to formulas (14) and (15)

\[ \hat{V}_k(x(k)) = W_{c2}\text{sig}(W_{c1} x(k) + b_1) \] (14)

\[ \hat{V}_{k+1}(x(k+1)) = W_{c2}\text{sig}(W_{c1} x(k+1) + b_1) \] (15)

where \(X(k) = [x^T(k), u^T(k)]^T, X(k+1) = [x^T(k+1), u^T(k+1)]^T\), and \(b_1\) is the activated threshold.

Formulas (14) and (15) transform the solving of \(V_k\) and \(V_{k+1}\) to the learning of connected weights \(W_{c1}\) and \(W_{c2}\). The backpropagation method takes an effect on training the parameters. The error \(\delta_c(k)\) between the estimated \(\hat{V}_k(x)\) and real \(V_k(x)\) is obtained by replacing \(V_k(x)\) according to formula (7) and \(\hat{V}_{k+1}(x(k+1))\) instead of \(V_{k+1}(x(k+1))\). Then we have

\[ \delta_c(k) = \hat{V}_k(x(k)) - V_k(x(k)) = \hat{V}_k(x(k)) - [R_k(x(k+1), x(k), u(k)) + \gamma V_{k+1}(x(k+1))] \]
Substituting formula (26) into the formulas (24) and (25), one gets

\[ \Delta W_{c2} = l \cdot W_{c2} \cdot \left[ s_{\text{out,c}} \left( 1 - s_{\text{out,c}} \right) \right] \cdot W_{c1,u} \cdot s_{\text{out,a}} \cdot x(k) \] (27)

\[ \Delta W_{a1} = l \cdot W_{c2} \cdot \left[ s_{\text{out,c}} \left( 1 - s_{\text{out,c}} \right) \right] \cdot W_{c1,u} \cdot W_{a2} \cdot \left[ s_{\text{out,a}} \left( 1 - s_{\text{out,a}} \right) \right] \cdot x(k) \] (28)

where \( W_{c1,u} \) and \( W_{c2} \) are the connected weight between \( u \) (part of \( X \)) and the hidden layer, and the connected weight between the hidden layer and output layer in critic network. \( s_{\text{out,c}} \) and \( s_{\text{out,a}} \) are the outputs of the sigmoid function in critic network and in action network, respectively; \( W_{a2} \) is the connected weight between the hidden layer and output layer in action network, and \( x(k) \) is the input of the action network, \( W_{c1,u}, W_{c2}, s_{\text{out,c}}, s_{\text{out,a}}, W_{a2} \) can be obtained from the critic network and action network.

The healthy model that is offline built based on the healthy data series will work parallel to the plant, and its dynamic performance is consistent with that of the plant under fault-free scenario. The well-trained feedforward neural network (FNN) can be used as a black-box healthy model [24]. The healthy model provides a current reference \( x_i(k + 1) \) which develops a target value for the critic action in the training process.

The optimization of the control variable will be obtained by adjusting the weight parameters of the action network based on the formulas (22) and (23). After each action selection, the critic network evaluates the result to determine whether it is better or worse than expected. By iterating the process of the critic network training and action network training, the connected weights of the critic network and action network will be convergent according to lemma 1. Furtherly one will get the optimal control \( u^*(k) \) responding to \( x(k) \) according to action network.

IV. IMPROVED BY PSO

The training of critic action architecture needs the alternation between the critic network and the action network in order to converge. It is a slow evolution progress that spends too much time in the process of trials and errors. In fact there are some efficient ways for animals after a long term evolution to find a goal in the unknown environment. One of them is particle swarm optimization (PSO) proposed by Kennedy and Eberhart [31] that provides a quick random searching approach by imitating the social behavior of the bird flock without depending on the model [32]. The iteration of the basic PSO follows formulas (29) and (30)

\[ v_i(k + 1) = \omega v_i(k) + c_1 \text{rand1} \cdot (p_{\text{best}} - x_i(k)) \]

\[ + c_2 \text{rand2} \cdot (g_{\text{best}} - x_i(k)) \] (29)

\[ x_i(k + 1) = x_i(k) + v_i(k) \] (30)

where \( \omega \) is an inertia weight, \( c_1 \) and \( c_2 \) are accelerating constants, \( \text{rand1} \) and \( \text{rand2} \) are two random numbers independently generated between \([0, 1]\), \( p_{\text{best}} \) is the best location of current particle swarm, and \( g_{\text{best}} \) is the best location of entire particle swarm experienced.

It is worth noting that the iteration of PSO is different from that of RL. The RL often gets the result for a long time until it is convergent meanwhile the PSO can get a solution after a few iterations due to a lot of particles’ parallel work.
A. FITNESS FUNCTION

The fitness function (FF) of PSO, as an evaluation of particles position, is identified as the similar form of temporal-difference error (TE) by replacing $V_k(x)$ and $V_{k+1}(x(k+1))$ with $\hat{V}_k(x)$ and $\hat{V}_{k+1}(x(k+1))$, respectively

$$\text{FF}(x(k)) = \hat{V}_k(x(k)) - R_k(x(k+1), x(k), u(k)) - \gamma \hat{V}_{k+1}(x(k+1)) = W_{c2}\text{sig}(W_{c1}x(k)) - R_k(x(k+1), x(k), u(k)) - \gamma W_{c2}\text{sig}(W_{c1}x(k+1))$$

(31)

where $\hat{V}_k(x)$ and $\hat{V}_{k+1}(x(k+1))$ are obtained from the critic network, the activated threshold is set as 0, and $R_k(x(k+1), x(k), u(k))$ is determined as the immediate cost error between the fault and the healthy

$R_k(x(k+1), x(k), u(k)) = x^T(k+1)Mx(k+1) - x^T(k+1)Mx(k+1)$

(32)

where $x(k+1)$ is the state at $k+1$ after $u$ and $x(k+1)$ is the states at $k+1$ from healthy model which is the target of FTC.

As a result, the seeking parameters $W_{c1}$ and $W_{c2}$ of the critic network becomes an optimal problem as follows:

$$\begin{align*}
\text{Min}[W_{c2}\text{sig}(W_{c1}[x^T(k)u^T]) - x^T(k+1)Mx(k+1) + x^T(k+1)Mx_k(k+1) - \gamma W_{c2} \\
\times \text{sig}(W_{c1}[x^T(k+1)u^T])]
\end{align*}$$

subject to $u = \text{output of action network}$

(33)

B. LEVEL SEARCH

The PSO provides an effective solution within a few iterations but it is prone to trap into local optimization [32], [33]. Here a level search approach is proposed to overcome it.

Suppose the local optimizations are divided to different levels according to the fitness function value. The particles begin to search the solution space from initial level and they are easy to trap into a local optimization. It is difficult for a standard PSO to escape it because the particles lost the searching ability due to the homoplasy at the local optimization. To overcome it a secondary search by changing the fitness function value and reassigning the particles searching ability with re-randomizing restarts a new search from this level. This search works until it traps into another local optimization and replaces by another new search from a new level. This process is going on until it finds a best fitness function value. The algorithm is summarized as procedure 1.

$$\begin{align*}
\text{Min}[W_{c2}\text{sig}(W_{c1}[x^T(k)u^T]) - x^T(k+1)Mx(k+1) + x^T(k+1)Mx_k(k+1) - \gamma W_{c2} \\
\times \text{sig}(W_{c1}[x^T(k+1)u^T])]
\end{align*}$$

subject to : keep $W_{c1}, W_{c2}, x(k), x(k+1)$ and $x_k(k+1)$ unchanged

(34)

Procedure 1

- Step 1: Initialize the particles responding to $W_{c1}$ and $W_{c2}$ of critic network
- Step 2: Compute the fitness function of each particle according to formula (31)
- Step 3: Renew the $p_{best}$ and $g_{best}$ according to

  $$p_{best} = \min\{FF_i(x_i(k)) | i = 1, 2, \cdots, l\}$$

  $$g_{best} = \min\{FF_i(x_i(k)) | i = 1, 2, \cdots, k\}$$

- Step 4: Update $v_i$ and $x_i$ according to formulas (29) and (30)
- Step 5: Repeat steps 2 to 4 until it is convergent and record $g_{best1}$
- Step 6: Modify the fitness function value and reassign the particles with random number [0, 1]
- Step 7: Repeat steps 2 to 4 until it is convergent and record $g_{best2}$
- Step 8: Repeat steps 2 to 7 until it cannot find a better solution $g_{best1} \geq g_{best2}$.
- Step 9: The position of particle at $g_{best1}$ is the solution of $W_{c1}$ and $W_{c2}$ of critic network.
- Step 10: Get a virtual optimal control $u^*(k)$ by solving an optimization of (34) using PSO again

It is noted that formula (33) and formula (34) have the same goal but different restrains. Formula (33) implies an achievable goal under the real control vector $u$ (the $u$ acts on the system at $k$ therein we get the state $x(k+1)$) and memorize this goal to the parameters $W_{c1}, W_{c2}$. Formula (34) provides an approach to get the optimal control $u^*(k)$ in order to catch this goal but this control $u^*(k)$ cannot be put into practice because the system has gone into the next time $k+1$. It is the reason that the $u^*(k)$ is a virtual optimal control. As stated above this $u^*(k)$ depends on the $x(k)$ and $x(k+1)$ whatever the faults happen. So the black box relations between $x(k)$ and $u^*(k)$ ($k = 1, 2, \cdots, n$) will be obtained by training the action network.

C. PSO IMPROVED FTC

The state $x(k)$ and the virtual optimal control $u^*(k)$ construct a sample element in training set which is consistent with the current fault. A training set is obtained after many attempts by changing states with the effect of different control actions. The training set for supervisory learning has been solved and the existing approach such as Levenberg-Marquardt algorithm [38] may play a significant role to improve the training efficiency. The schematic block-diagram of the FTC controller improved by PSO is dictated in figure 2 and the algorithm is summarized as procedure 2.

Remark 2: The virtual optimal control obtained by PSO level search makes it possible to build a training dataset which makes the RL training become a supervised learning. This provides a higher training efficiency instead of the slow trials. Meanwhile the advantage of CAFTC with no model, no fault style and no prior fault information is inherited due to the data of training set directed from the system subject to fault.
V. CASE STUDY

A three-tank system is implemented as a testbed to verify the proposed approach. The three-tank system is made up with tank body (T1, T2 and T3), submersible pumps (pump1 and pump2) whose flow (Q1,Q2) is controlled by a digital controller, the connection valves (CV1,CV2,CV3), the leak valves (LV1,LV2,LV3) and pipes. The liquid height of each tank (T1,T2 and T3) can be obtained respectively by a pressure level meter. Three tanks (T1,T2 and T3) have the same size connected with pipes. The system works in the condition of opening connection valves and closed leak valves. Therefore, the liquid of the tank body flows to the reservoir through CV3 and reenters the tank body by pump1 and pump2. The character of the plant can be changed with the flow resistance between tanks by manual regulating the opening of connection valves and leak valves. The pump1 and pump2 are controlled by an individual frequency converter. The flux of pump1 and pump2 is determined by their rotation speed that is controlled by an individual frequency converter. The control signal 0-5V of frequency converter is taken as the output of controller. The relation between the flux of pump and the control signal of frequency is obtained in advance by additional experiments. Afterwards we take the flux of pumps instead of rotation speed as the control variable of plant by omitting the frequency converter in order to be clear. The structure is shown in Fig. 3.

\[
\begin{align*}
\text{FIGURE 3.} & \quad \text{The structure of three-tank system.} \\
\end{align*}
\]

The model of three tank system is given as formula (35)

\[
\begin{align*}
\frac{dh_1}{dt} &= -\frac{\lambda_{13}}{S_1} \cdot \text{sgn}(h_1 - h_3) \sqrt{2g(h_1 - h_3)} + \frac{Q_1}{S_1} \\
\frac{dh_2}{dt} &= -\frac{\lambda_{32}}{S_2} \cdot \text{sgn}(h_2 - h_3) \sqrt{2g(h_2 - h_3)} - \frac{\lambda_2}{S_2} \\
\frac{dh_3}{dt} &= -\frac{\lambda_{13}}{S_3} \cdot \text{sgn}(h_1 - h_3) \sqrt{2g(h_1 - h_3)} - \frac{\lambda_{32}}{S_3} \\
&\quad \cdot \text{sgn}(h_3 - h_2) \sqrt{2g(h_3 - h_2)} \\
\end{align*}
\]

where \(h_1\), \(h_2\) and \(h_3\) are the liquid heights of T1,T2 and T3, \(S_i\) is the base area of T1, \(\lambda_{13} = \frac{R_{13}}{\rho g}, \lambda_{32} = \frac{R_{32}}{\rho g}, \lambda_2 = \frac{R_2}{\rho g}\), \(R_{13}, R_{32}\) is the flow resistance between tank i and j, \(R_2\) is the drainage resistance of CV3, \(R_{13}, R_{32}\) and \(R_2\) are obtain by adjusting the opening of CV1, CV2 and CV3, \(\rho\) is the liquid density (for water, \(1 \times 10^3\)kg/m\(^3\), \text{sgn}(*)\) is the sign function with \(\text{sgn}(*) = \begin{cases} 1 & if \ast > 0 \\ 0 & if \ast = 0 \\ -1 & if \ast < 0 \end{cases}\), \(g\) is the acceleration of gravity, \(Q_1\) and \(Q_2\) is the flow from pump1 and pump2.

We apply a PID controller for pump1 and keep pump2 at the 50% opening (mid signal 2.5V of control signal 0-5V of frequency converter from soft way) to implement our target of keeping the liquid level in T3 in the condition of fault free. We call this stability subject to fault free as a standard state. When the fault occurs a FTC controller with two outputs (the flux of pump1 and pump2) is to replace the former controller (a PID for pump1 and 50% fixed opening for pump2). Our target is to preserve a reference liquid level in T3 by controlling the flow of pump1 and pump2 respective.
A. SCENARIO OF ACTUATOR OUTPUT DEVIATION FAULT

An actuator fault of pump 1 is imitated by changing the relation between the flux of pump and the control signal of frequency subject to fault free scenario. This change makes the flux add/lessen compared with the initial set that connects the output of controller. By this means the actuator output deviation fault is obtained with soft way which prevents the real actuator from damaging. An actuator fault of pump 1 which pumps 12 l/min (be translated according to the relation between the flux of pump and the control signal of frequency) more fluid than its initial set is added after sampling 100. The liquid height of T3, the states evolution and the control variables are shown in figure 4, figure 5 and figure 6.

B. SCENARIO OF ACTUATOR STUCK FAULT

A stuck fault for pump 1 at 60% opening (the signal 3V of control signal 0-5V of frequency converter, means the pump1 is stroked due to lose the control) is given after sampling 100. Figure 7, figure 8 and figure 9 are the liquid height of T3, the states evolution and the control variables respective.

The blue curve and the red curve represent the cases without FTC and with FTC. Figure 5 shows that the states x1, x2 and x3 under fault free condition are keeping stable before the fault occurrence and the liquid height of T3 is maintained in the reference level (figure 4). When the fault occurs at sample 100 the state x1 will go up because of the more flow of T1 and the states x2, x3 will also go up due to the coupling in the case of without FTC. But it will go into another stability for the liquid height of T3 from 10cm to 15cm after a transient process. A FTC controller is designed according to Procedure 1 with a result of a forward neural network with 3-10-2. The training set is selected with 100 data and trained with Levenberg-Marquardt algorithm. The well trained NN is used as FTC. It is obvious that the liquid height of T3 is recovered due to the proposed algorithm.

More explanations about control variables will be given on the base of figure 6. In figure 6 the horizontal ordinate is sampling time and the longitudinal is the flux of pump. The scale zero of longitudinal coordinates represents the flux of pumps in the standard state. We use scale zero instead of the real flux because the standard state will vary with the reference level of T3 subject to fault free. Negative means less flow and positive means more flow than the standard state subject to fault free. The red curve and the blue curve represent the flux without FTC and with FTC respectively. It is seen that the pump 1 will reduce the flow to react on the actuator more output fault. On the other hand the pump 2 will also reduce the output in order to keep the T3 liquid height at the reference level.
variables with FTC (red curve) and without FTC (blue curve). The red curve and the blue curve are coincided because the pump1 is jammed and loses its regulating function. The pump2 reflects this fault by stopping delivering the flow for a while in order to release the accumulation. Then it will provide the stable flow to maintain the level of T3. The reason of no difference between the blue curve and the red curve after fault is due to the error that is dictated at the following paragraph. The figure 7 shows the liquid height of T3 can maintain the height subjected to fault free (red curve) with the control of FTC.

The similar stuck fault with pump 1 opening reduced to 30% (the signal 1.5V of control signal 0-5V of frequency converter) that cannot maintain the liquid level go up. The states evolution, the liquid height of T3 and the control variables are shown as figure 11, figure 10 and figure 12. The blue curves present the cases without FTC and the red curves show the states evolution with FTC. It is seen from figure 12 that there is a lower intensity and shorter time of releasing flow compared with the 60% opening stuck fault. It also appears a deviation between the red curve and the blue curve due to the difference of stabilities.

C. SCENARIO OF PLANT LEAK FAULT

We also made a flow leak fault by partly opening LV2 of tank T3. It is dictated from figure 13 the liquid height in T3 will reduce from 9cm to 7cm due to the flow leak if the control subject to fault free is implementing which is shown in the blue curve. The red curve of figure 13 shows the tendency of liquid height in T3 under the proposed FTC. One can see the liquid height in T3 will hold a level subject to fault free.
with contribution of FTC. The states evolution and the control variables are seen in figure 14 and figure 15.

![Figure 14. The states evolution.](image)

![Figure 15. The control variables.](image)

**D. SOME CONCERNS**

To our best knowledge there is no other data-driven FTC to deal with all kinds of faults without any preconditions except the redundancy of system. This approach should go into effect as soon as the fault is detected without further diagnosis. It is relatively easy in FDD and there are many mature methods such as PCA, Bayesian decision and so forth. What is more, the critic action architecture has an excellent robustness due to its approaching with neural network which will resist the disturbance of noise.

The transient time from fault-detection to control-action depends on the number of samples and their training time under supervised training which is less than that of traditional trials and errors. More details of shorten time is seen in our early work [39]. In this paper the transient process has been omitted in order to highlight the effect of proposed FTC.

**VI. CONCLUSION**

A critic action architecture in RL has been proposed to design a fault tolerant controller to overcome the difficulty of the unknown changes in a plant due to the fault suddenness and unpredictability. The PSO using level search will find the virtual optimal control on the current states without training dataset and then construct a training dataset of the supervisory training which is more effective in training than the conventional RL. The proposed approach is a data-driven FTC without any prior knowledge of system that avoids the bottleneck problem of modeling. The FTC has been obtained by a training process using the online data that reliably reflect the fault process. Like other data-driven methods the proposed approach needs the collected data as an analysis basis. It has an ability of getting an optimal action target on the performance indicator under the condition of unexpected fault.

**REFERENCES**

[1] Y. Zhang and J. Jiang, “Bibliographical review on reconfigurable fault-tolerant control systems,” *Annu. Rev. Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008.

[2] S. Yin, B. Xiao, S. X. Ding, and D. Zhou, “A review on recent development of spacecraft attitude fault tolerant control system,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3311–3320, May 2016.

[3] H. Habibi, I. Howard, and S. Simani, “Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review,” *Renew. Energy*, vol. 135, pp. 877–896, May 2019.

[4] F. Bateman, H. Noura, and M. Ouladsine, “Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 2119–2137, Jul. 2011.

[5] Z. Chen, X. Li, C. Yang, T. Peng, C. Yang, H. R. Karimi, and W. Gui, “A data-driven ground fault detection and isolation method for main circuit in railway electrical traction system,” *IJA Trans.*, vol. 87, pp. 264–271, Apr. 2019.

[6] X. Yu and J. Jiang, “A survey of fault-tolerant controllers based on safety-related issues,” *Annu. Rev. Control*, vol. 39, pp. 46–57, Jan. 2015.

[7] S. Yin, H. Luo, and S. X. Ding, “Real-time implementation of fault-tolerant control systems with performance optimization,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May 2014.

[8] Z. Guo, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.

[9] H. Wang, P. X. Liu, X. Zhao, and X. Liu, “Adaptive fuzzy finite-time control of nonlinear systems with actuator faults,” *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1786–1797, May 2020.

[10] S. H. Gao, Y. W. Jing, X. P. Liu, and S. Y. Zhang, “Finite-time attitude-tracking control for rigid spacecraft with actuator failures and saturation constraints,” *Int. J. Robust Nonlinear Control*, vol. 30, no. 5, pp. 1903–1937, Mar. 2020.

[11] S. X. Ding, Y. Wang, S. Yin, P. Zhang, Y. Yang, and E. L. Ding, “Data-driven design of fault-tolerant control systems,” in *Proc. 8th IFAC Symp. Fault Detection*, Mexico City, Mexico, Aug. 2012, pp. 1323–1328.

[12] Z. Gao, H. Saxen, and C. Gao, “Data-driven approaches for complex industrial systems,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2210–2212, Nov. 2013.

[13] S. X. Ding, “Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results,” *J. Process Control*, vol. 24, no. 2, pp. 431–449, Feb. 2014.

[14] J. Wang and G. Yang, “Data-driven output-feedback fault-tolerant L2 control of unknown dynamic systems,” *ISA Trans.*, vol. 63, pp. 182–195, Jul. 2016.

[15] Z. Wang, L. Liu, and H. Zhang, “Neural network-based model-free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2351–2362, Aug. 2017.

[16] X. Mao, Y. Wang, and X. Liu, “A hybrid feedforward-feedback hysteresis compensator in piezoelectric actuators based on least-squares support vector machine,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 54–5711, Jul. 2018.

[17] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, “Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks,” *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 101–110, Feb. 2018.

[18] Z. Guo, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.
D. Zhang, Z. Gao: FTC Using Reinforcement Learning and PSO

[19] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” J. Artif. Intell. Res., vol. 4, no. 1, pp. 237–285, Jan. 1996.

[20] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 2005.

[21] L. Liu, Z. Wang, and H. Zhang, “Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters,” IEEE Trans. Autom. Sci. Eng., vol. 14, no. 1, pp. 299–313, Jan. 2017.

[22] Z. Wang, L. Liu, H. Zhang, and G. Xiao, “Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 5, pp. 22–611, May 2016.

[23] D. Zhang, Z. Lin, and Z. Gao, “Reinforcement-learning based fault-tolerant control,” in Proc. IEEE 15th Int. Conf. Ind. Informat. (INDIN), Jul. 2017, pp. 671–676.

[24] D. Zhang and Z. Gao, “Reinforcement learning–based fault-tolerant control with application to flux cored wire system,” Meas. Control, vol. 51, nos. 7–8, pp. 349–359, Sep. 2018.

[25] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Proc. 33rd Int. Conf. Mach. Learn. (ICML), vol. 4, 2016, pp. 2850–2869.

[26] S. Gu, T. Lillicrap, U. Sutskever, and S. Levine, “Continuous deep q-learning with model-based acceleration,” in Proc. 33rd Int. Conf. Mach. Learn. (ICML), vol. 6, 2016, pp. 4135–4148.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in Proc. 32nd Int. Conf. Mach. Learn. (ICML), vol. 3, 2015, pp. 1889–1897.

[28] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, “Sample efficient actor-critic with experience replay,” Nov. 2016, arXiv:1611.01224. [Online]. Available: http://arxiv.org/abs/1611.01224

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” Jul. 2017, arXiv:1707.06347. [Online]. Available: http://arxiv.org/abs/1707.06347

[30] H. Cho, P. Oh, J. Park, W. Jung, and J. Lee, “FA3C: FPGA-accelerated deep reinforcement learning,” in Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst. (ASPLOS), Apr. 2019, pp. 499–513.

[31] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE ICNN, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[32] K. Kameyama, “Particle swarm optimization—a survey,” IEICE Trans. Inf. Syst., vol. E92D, no. 7, pp. 1354–1361, Jul. 2009.

[33] A. Rezaee Jordehi, “Particle swarm optimization for dynamic optimisation problems: A review,” Neural Comput. Appl., vol. 25, nos. 7–8, pp. 1507–1516, Dec. 2014.

[34] P. J. Werbos, “Neural networks for control and system identification,” Heuristics, vol. 3, no. 1, pp. 18–27, Spring 1990.

[35] D. V. Prokhorov and D. C. Wunsch, “Adaptive critic designs,” IEEE Trans. Neural Netw., vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[36] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, “Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming,” Automatica, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[37] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, “Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 38, no. 4, pp. 943–949, Aug. 2008.

[38] L. S. H. Ngia and J. Sjoberg, “Efficient training of neural nets for nonlin-ear adaptive filtering using a recursive Levenberg-Marquardt algorithm,” IEEE Trans. Signal Process., vol. 48, no. 7, pp. 1915–1927, Jul. 2000.

[39] D. Zhang, H. Zhang, C. Li, and A. Wu, “Particle swarm optimized reinforcement learning-based fault tolerant control,” in Proc. 15th World Congr. Intel1l. Control Autom. (WCICA), Jul. 2018, pp. 515–519.

DAPENG ZHANG received the Ph.D. degree in control science and engineering from Northeastern University, China. He is currently a Lecturer with the School of Electrical and Information Engineering, Tianjin University, China. His research interests include reinforcement learning, industrial control, fault detection and diagnosis, and fault tolerant control.

ZHIWEI GAO (Senior Member, IEEE) received the B.Eng. degree in electric engineering and automation, and the M.Eng. and Ph.D. degrees in systems engineering from Tianjin University, in 1987, 1993, and 1996, respectively.

He currently works with the Faculty of Engineering and Environment, University of Northumbria, Newcastle upon Tyne, U.K., as a Reader. His research interests include data-driven approaches, machine learning, estimation and filtering, fault diagnosis, resilient control, intelligent optimization, power electronics, wind energy systems, electrical vehicle batteries, and bioinformatics. He was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE ACCESS, and ISA TRANSACTIONS, and an Editorial Member of Renewable Energy (Elsevier).