

Northumbria Research Link

Citation: Zhang, Dapeng and Gao, Zhiwei (2018) Reinforcement learning-based fault-tolerant control with application to flux cored wire system. *Measurement and Control*, 51 (7-8). pp. 349-359. ISSN 0020-2940

Published by: SAGE

URL: <http://dx.doi.org/10.1177/0020294018789202>
<<http://dx.doi.org/10.1177/0020294018789202>>

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/36162/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



**Northumbria
University**
NEWCASTLE



UniversityLibrary

Reinforcement learning-based fault-tolerant control with application to flux cored wire system

Measurement and Control
2018, Vol. 51(7-8) 349–359
© The Author(s) 2018
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0020294018789202
journals.sagepub.com/home/mac


Dapeng Zhang¹  and Zhiwei Gao²

Abstract

Background: Processes and systems are always subjected to faults or malfunctions due to age or unexpected events, which would degrade the operation performance and even lead to operation failure. Therefore, it is motivated to develop fault-tolerant control strategy so that the system can operate with tolerated performance degradation.

Methods: In this paper, a reinforcement learning-based fault-tolerant control method is proposed without need of the system model and the information of faults.

Results and Conclusions: Under the real-time tolerant control, the dynamic system can achieve performance tolerance against unexpected actuator or sensor faults. The effectiveness of the algorithm is demonstrated and validated by the rolling system in a test bed of the flux cored wire.

Keywords

Fault-tolerant control, reinforcement learning, performance index, flux cored wire

Date received: 12 April 2018; accepted: 13 June 2018

Introduction

With high demands on reliability, safety, and availability in industrial automation systems, fault-tolerant control (FTC) has been an important research topic during the last four decades. An early holistic view of FTC was given in Blanke et al.,¹ and recent surveys and overviews were documented in Gao et al.,² Yu and Jiang,³ Zhang and Jiang,⁴ and Yin et al.⁵ FTC techniques are divided into passive FTC and active FTC. Compared with the passive FTC, the capabilities of which diminish as the number of fault scenarios increases, the active FTC is more flexible to deal with different types of faults.⁶ Fault diagnosis methods can be categorized into model-, signal-, and knowledge-based methods.² As a possible solution to ensure safe and reliable operation of the system, the model-based fault detection and isolation (FDI) and FTC techniques have achieved fruitful results.^{7–12} The sliding mode control,⁸ adaptive decentralized control,⁹ and coprime factorization techniques¹⁰ have been applied successfully. Youla parameterization-based FTC was developed in Ding et al.¹¹ and Yin et al.¹² to solve a nonlinear system.

Modern automation industries enable the availability of a large amount of historical data. As a result, data-driven modeling, diagnosis, and FTC have become a hot research topic. Data can be used for learning to extract the knowledge base

such as using fuzzy approximation,¹³ neural network (NN)-based methods,¹⁴ K-clustering,¹⁵ and support vector machines (SVM).^{16,17} Moreover, fault feature can also be exploited from the collected data, such as using principal component analysis (PCA),^{18,19} empirical mode decomposition (EMD),^{20,21} and so forth.^{22–25} It is the performance indicators (PIs) that are very important in the industrial process. Both sensors and control system will target on the plant to obtain the benefits of maximization by optimizing the performance indicator within the scope of safety. FTC is a good alternative that has the capability of approaching to performance indicator without any fault by adjusting the system variables. In Yin et al.,¹² a gradient-based optimization method was given to optimize the system performance by means of disturbance rejection. In Macgregor and Cinar,²² a recursive total principle component regression (R-TPCR)-based

¹School of Electrical and Information Engineering, Tianjin University, Tianjin, China

²Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, UK

Corresponding author:

Dapeng Zhang, School of Electrical and Information Engineering, Tianjin University, No. 92 Weijian Road, Naikai District, Tianjin 300072, China.
Email: zdp@tju.edu.cn



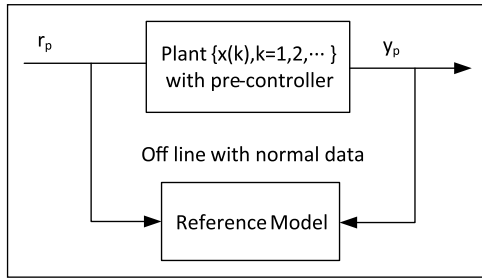


Figure 1. Obtaining the identification for the reference model.

design and implementation approach was proposed for efficient data-driven FTC and optimization.

Strongly motivated by keeping the performance indicator fault free, it is of interest to seek an FTC approach in order to preserve the normal performance indicator under all kinds of unexpected fault scenarios. The performance indicator without any fault is a reflection of the system's real ability and easy to be gained from the obtained data. However, it becomes a challenge in the case of fault because the unexpected fault has changed the maps of system states and made the original controller unavailable; meanwhile, there are not enough valid data to develop an FTC controller for an early fault. Reinforcement learning provides an inspiration to solve the above problem. Reinforcement learning is about learning from the interaction on how to behave in order to achieve a goal.²⁶⁻²⁸ The reinforcement learning agent and its environment interact over a sequence of discrete time steps and gain a series of optimal actions finally. If an unexpected fault is considered as the environment, and the performance of the system under fault-free condition is regarded as the desired goal, the controller can be designed by reinforcement learning to achieve the optimal behavior. In this paper, a novel structure of FTC is proposed based on reinforcement learning, and the advantages are given as follows:

1. This approach is a data-driven method without knowing the mechanism model of plant;
2. This approach is an online method which is suitable for unexpected faults without prior fault information;
3. The controller has the ability to take optimal actions to mitigate the adverse influences from faults.

Problem description and preliminaries

Problem description

Suppose a time series of the plant $\{x_p(k), k=1,2,\dots,\infty\}$ with inputs $\{r_p(k), k=1,2,\dots,\infty\}$ and outputs $\{y_p(k), k=1,2,\dots,\infty\}$ where the dynamics of the plant is stable using a pre-designed controller. In addition, the system states are assumed to be measurable. We use a descriptive definition instead of mathematical expression in order to highlight the data of time series without considering any additional parameters.

The addressed FTC system is composed of three units: the plant, the reference model, and the fault-tolerant controller. The plant works well under fault-free condition with a pre-designed controller, which is represented as a form of time series of the data $\{x_p(k), k=1,2,\dots,\infty\}$. A reference model is used to provide information of states under fault-free condition, and the reference model is built based on the healthy data series that will work parallel to the plant, and its dynamic performance is consistent with that of the plant under fault-free scenario. The FTC will produce a control variable based on the real-time system states of the plant and the reference model. Their relations will be discussed below.

Reference model

For a system, the reference model can be expressed in the form of

$$x_m(k+1) = f(x_m(k)) \quad (1)$$

where $x_m = (x_{m,1}, x_{m,2}, \dots, x_{m,n})^T \in \mathcal{R}^n$ is the state vector of the reference model and $f(\cdot)$ is the state transition function obtained from the time series of the plant.

There are various methods for system identification such as least square method (LSM), maximum likelihood method (MLM), and NN. A reference model can be obtained using the data of the time series of the plant under fault-free condition, as depicted in Figure 1.

Performance index

The stage costs J_p of the plant are given as

$$J_p = \sum_{k=1}^{\infty} [x_p^T(k) M x_p(k) + r_p^T(k) N r_p(k)] \quad (2)$$

where $x_p \in \mathcal{R}^n$ and $r_p \in \mathcal{R}^m$ are the state vector and the reference input of the plant, respectively, and M and N are the selected weighted matrices.

The stage costs J_m of the reference model are given as

$$J_m = \sum_{k=1}^{\infty} [x_m^T(k) M x_m(k) + r^T(k) N r(k)] \quad (3)$$

where $x_m \in \mathcal{R}^n$ and $r_m \in \mathcal{R}^m$ are the state vector and the reference input of the reference model, respectively.

Define the performance index $V(x_p, x_m)$ as the error in stage costs between the plant and the reference model, that is, $V(x_p, x_m) = J_p - J_m$. It is the goal of reinforcement learning sub-section D. Notice that $r_p = r$, so we get the performance index $V(x_p, x_m)$ of stage

$$V(x_p, x_m) = \sum_{k=1}^{\infty} [x_p^T(k) M x_p(k) - x_m^T(k) M x_m(k)] \quad (4)$$

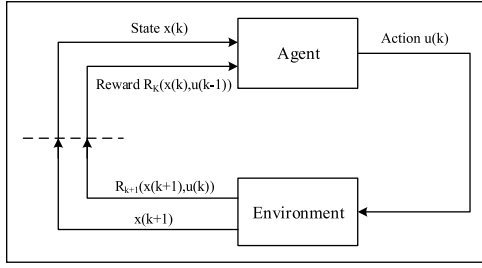


Figure 2. A basic frame of reinforcement learning.

and the performance index $V_k(x_p, x_m)$ of stage from time k

$$V_k(x_p, x_m) = \sum_{i=k}^{\infty} [x_p^T(i) M x_p(i) - x_m^T(i) M x_m(i)] \quad (5)$$

It is obvious when $x_p(k) = x_m(k)$, $V(x_p, x_m) = 0$, that is, the reference model and the real-time process have the same performance index under fault-free condition.

Remark 1

1. The stage costs J_p of the plant should be consistent with those of the reference model J_m under fault-free condition.
2. Under faulty condition, one can regulate the real-time states/outputs by tracking the reference model performance using learning algorithms.

Reinforcement learning method

In fact, it seems that we know little about real-time dynamics after the fault occurs. The traditional controllers often struggle to provide an effective control due to the lack of information on real-time dynamics. The reinforcement learning that is motivated by statistics, psychology, neuroscience, and computer science is a powerful tool to deal with uncertain surrounding by interacting with its environment. As in Bhatnagar and Babu,²⁹ Watkins and Dayan,³⁰ Sutton and Barto,³¹ Bradtke and Ydstie,³² and Ngia and Sjoberg,³³ the basic theory and methods of the reinforcement learning are simply introduced here. The basic frame of reinforcement learning is shown in Figure 2.²⁶

An agent will get the evaluation of good or bad behavior on the environment and learn through experience without a teacher who teaches how to do. In each training session, named episode, the agent explores the environment by changing action $u(k)$ and receives the state $x(k+1)$ and the immediate cost $R_{k+1}(x(k+1), u(k))$. The purpose of the training is to enhance the ‘brain’ of the agent. The goal of an agent is to minimize the immediate cost $\sum_{i=k}^{\infty} R_i$ which is received in the long run.

Consider a Markov decision process MDP $(\mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R})$, where \mathcal{X} is a set of states and \mathcal{U} is a set of actions or controls. The transition probabilities $\mathcal{P} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow [0, 1]$ represent for each state $x \in \mathcal{X}$ and action $u \in \mathcal{U}$ the conditional probability $P(x(k+1), x(k), u(k)) = \Pr\{x(k+1) | x(k), u(k)\}$ of transitioning to state $x(k+1) \in \mathcal{X}$ where the Markov decision process MDP is in state $x(k)$ and takes action

$u(k)$. The cost function $\mathcal{R} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{R}$ is the expected immediate cost $R_k(x(k+1), x(k), u(k))$ paid after transition to state $x(k+1) \in \mathcal{X}$, given that the Markov decision process MDP starts in state $x(k) \in \mathcal{X}$ and takes action $u(k) \in \mathcal{U}$.

The value of a policy $V_k^\pi(x(k))$ is defined as the conditional expected value of the future cost $E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} R_i \right\}$, $R_i \in \mathcal{R}$ when starting in state $x(k)$ at time k and following policy $\pi(x, u)$ thereafter

$$\begin{aligned} V_k^\pi(x) &= E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} R_i \right\} \\ &= \sum_u \pi(x, u) \sum_{x(k+1)} P \left(\begin{array}{c} x(k+1), \\ x(k), u(k) \end{array} \left[\begin{array}{c} R_k(x(k+1), x(k), u(k)) \\ + \gamma E_\pi \left\{ \sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} R_i \right\} \end{array} \right] \right) \\ &= \sum_u \pi(x, u) \sum_{x(k+1)} P \left(\begin{array}{c} x(k+1), \\ x(k), u(k) \end{array} \left[\begin{array}{c} R_k(x(k+1), x(k), u(k)) \\ + \gamma V_{k+1}^\pi(x(k+1)) \end{array} \right] \right) \end{aligned} \quad (6)$$

where $T = \infty$, $P(x(k+1), x(k), u(k))$ represents the transition probability from $x(k)$ to $x(k+1)$ under an action $u(k)$. For all the $x(k+1)$ starting in state $x(k)$ at time k , the whole transition probability following an action $u(k)$ is $\sum_{x(k+1)} P(x(k+1), x(k), u(k))$. The term $\pi(x, u) \sum_{x(k+1)} P(x(k+1), x(k), u(k))$ is defined as the transition probability following policy $\pi(x, u)$. For all the actions $u(k)$, the value function $V_k^\pi(x(k))$ for the policy $\pi(x(k), u(k))$ satisfies the Bellman equation

$$\begin{aligned} V_k^\pi(x(k)) &= \sum_u \pi \left(\begin{array}{c} x(k), \\ u(k) \end{array} \right) \sum_{x(k+1)} P(x(k+1), x(k), u(k)) \\ &\quad \left[\begin{array}{c} R_k(x(k+1), x(k), u(k)) \\ + \gamma V_{k+1}^\pi(x(k+1)) \end{array} \right] \end{aligned} \quad (7)$$

Therefore, the optimal actions can be gained by alternating the value iteration (equation (8)) and policy iteration (equation (9)) according to the following two equations

$$\begin{aligned} V_k(x(k)) &= \sum_u \pi_k \left(\begin{array}{c} x(k), \\ u(k) \end{array} \right) \sum_{x(k+1)} P(x(k+1), x(k), u(k)) \\ &\quad \left[\begin{array}{c} R_k(x(k+1), x(k), u(k)) \\ + \gamma V_k(x(k+1)) \end{array} \right] \end{aligned} \quad (8)$$

$$\begin{aligned} \pi_k \left(\begin{array}{c} x(k), \\ u(k) \end{array} \right) &= \arg \min_{\pi} \sum_{x(k+1)} P \left(\begin{array}{c} x(k+1), \\ x(k), u(k) \end{array} \right) \\ &\quad \left[\begin{array}{c} R_k(x(k+1), x(k), u(k)) \\ + \gamma V_k(x(k+1)) \end{array} \right] \end{aligned} \quad (9)$$

where γ is a discount factor with $0 < \gamma < 1$ in order to be convergent.

To a deterministic system $\sum_u \pi_k(x, u) \sum_{x(k+1)} P(x(k+1), x(k), u(k)) = 1$, the formulae (8) and (9) are written as

$$V_k(x(k)) = R_k(x(k+1), x(k), u(k)) + \gamma V_k(x(k+1)) \quad (10)$$

$$\pi_k(x(k), u(k)) = \arg \min_{\pi} R_k(x(k+1), x(k), u(k)) + \gamma V_k(x(k+1)) \quad (11)$$

There is only state information in formulae (10) and (11). One can obtain the optimal action only using the current state, but without knowing the system dynamics.

S Bradtke and BE Ydstie³² presented the stability and convergence results for dynamic programming-based reinforcement learning applied to linear quadratic regulation. Here, reinforcement learning is used to design a tolerant controller for systems subjected to faults.

Reinforcement learning-based FTC design

Reinforcement learning-based FTC

The first thing to apply in reinforcement learning is to determine the cost function weight at time k . Here one can define the cost function $R_k(x_{\Delta}(k))$ at time k as the quadratic form of $x_{\Delta}(k) = x_p(k) - x_m(k)$

$$R_k(x_{\Delta}(k)) = x_{\Delta}^T(k) M x_{\Delta}(k) \quad (12)$$

The function $V_k(x_{\Delta}(k))$ after time k is defined as

$$V_k(x_{\Delta}) = \sum_{i=k}^{\infty} \gamma^{i-k} R_i(x_{\Delta}(i)) \quad (13)$$

where γ is a discount factor, $0 < \gamma < 1$.

As a result, we have

$$V_k(x_{\Delta}(k)) = R_k(x_{\Delta}(k)) + \gamma V_{k+1}(x_{\Delta}(k+1)) \quad (14)$$

Following the Bellman optimal equation,²⁹⁻³¹ the optimal value function $V^*(x_{\Delta}(k))$ is obtained according to the following formula

$$V^*(x_{\Delta}(k)) = \min_{u(k)} \{ R_k(x_{\Delta}(k), u(k)) + \gamma V_{k+1}(x_{\Delta}(k+1)) \} \quad (15)$$

where $V^*(\cdot)$ and $u(k)$ are the optimal value function and the control variable at time k , respectively. $R_k(x_{\Delta}(k))$ in equation (14) is denoted as $R_k(x_{\Delta}(k), u(k))$ in equation (15) in order to highlight the effect of $u(k)$ because $R_k(x_{\Delta}(k))$ can be adjusted by changing $u(k)$.

It is noticed that equation (15) cannot be used online because one cannot know the cost function of the future time, that is, $V_{k+1}(x_{\Delta}(k+1))$. A Q -algorithm²⁹ provides an

effective solution by substituting function Q in equation (15).

The evaluation function $Q_k(x_{\Delta}(k), u(k))$ instead of $V_k(x_{\Delta}(k))$ is defined as the minimum discounted cumulative reward that can be achieved from state $x_{\Delta}(k)$ and $u(k)$ as the first action

$$Q_k(x_{\Delta}(k), u(k)) \stackrel{\text{def}}{=} R_k(x_{\Delta}(k), u(k)) + V^*(\delta(x_{\Delta}(k), u(k))) \quad (16)$$

where $V^*(\delta(x_{\Delta}(k), u(k)))$ is the optimal value function of $V_{k+1}(x_{\Delta}(k+1))$ and $\delta(x_{\Delta}(k), u(k))$ expresses the state $x_{\Delta}(k+1)$ that comes from $x_{\Delta}(k)$ and $u(k)$, that is, $x_{\Delta}(k+1) = \delta(x_{\Delta}(k), u(k))$. We denote $\delta(x_{\Delta}(k), u(k))$ in order to stress the relation between $x_{\Delta}(k+1)$ and $x_{\Delta}(k)$ and $u(k)$.

If $Q_k(x_{\Delta}(k), u(k))$ achieves its optimization under some action $u(k)$, the function $V_k(x_{\Delta}(k), u(k))$ can also achieve its optimization with the same action. As a result, $V_k(x_{\Delta}(k), u(k))$ may be replaced by $Q_k(x_{\Delta}(k), u(k))$. This implicates that the optimal action can be obtained only by the evaluation function $Q_k(x_{\Delta}(k), u(k))$ without using the value function $V_k(x_{\Delta}(k), u(k))$.

Denote the optimum of $Q_k(x_{\Delta}(k), u(k))$ as $Q_k^*(x_{\Delta}(k), u(k))$; therefore, one has

$$\begin{aligned} Q_k^*(x_{\Delta}(k), u(k)) &= \min_{u(k)} [R_k(x_{\Delta}(k), u(k)) \\ &\quad + V^*(\delta(x_{\Delta}(k), u(k)))] \\ &= R_k^*(x_{\Delta}(k), u(k)) \\ &\quad + V^*(x_{\Delta}(k+1)) \\ &= V_k^*(x_{\Delta}(k), u(k)) \end{aligned} \quad (17)$$

where the superscript * expresses the optimal values.

It is seen from formula (17) that $Q_k^*(x_{\Delta}(k), u(k))$ is equivalent to $V_k^*(x_{\Delta}(k), u(k))$ with the same action. Therefore, the value iteration formula (10) of a deterministic system is transformed to a form of Q according to $x_{\Delta}(k)$ and a fixed $u(k)$

$$Q_k(x_{\Delta}(k), u(k)) = R_k(x_{\Delta}(k), u(k)) + \gamma Q_{k+1}(x_{\Delta}(k+1), u(k+1)) \quad (18)$$

The policy iteration formula (11) of a deterministic system is transformed to formula (19)

$$\pi_k(x_{\Delta}(k), u(k)) = \arg \min_{u(k)} Q_k(x_{\Delta}(k), u(k)) \quad (19)$$

The optimal action $\pi_k^*(x_{\Delta}(k), u(k))$ can be solved by alternating with two procedures: policy evaluation and policy improvement.

The policy evaluation is to find the optimal value in the case of current policy $\pi_k(x(k), u(k))$ by iteration. Select an initial policy $\pi_0(x(k), u(k))$ randomly. Denote j as the number of iterations and starting with $j=0$, iterate on j according to formula (20) until convergence

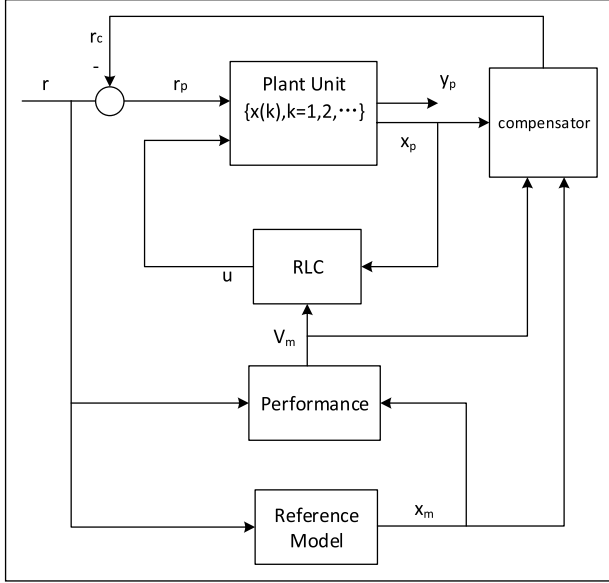


Figure 3. The schematic block diagram with compensator.

$$Q_k^{(j+1)}(x_\Delta(k), u(k)) = R_k(x_\Delta(k), u(k)) + \gamma \min_{u(k+1)} Q_k^{(j)}(x_\Delta(k+1), u(k+1)) \quad (20)$$

The policy improvement is to find another policy that is better, or at least no worse according to equation (19) by greedy method.

The alternation procedures are given as follows³¹

$$\pi_0 \xrightarrow{E} Q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} Q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi^* \xrightarrow{E} Q^*$$

where E and I are the policy evaluation and policy improvement, π_0 and Q_{π_0} are the initial policy and Q value, π_j and Q_{π_j} are the policy and Q value at the iteration time j , and π^* and Q^* are the end policy and Q value, respectively.

It is important for the iteration to be convergent, and the convergence of the Q learning for deterministic Markov decision process (MDP) is given in Sutton and Barto,³¹ shown as follows:

Lemma 1. Consider a Q learning agent in a deterministic Markov decision process MDP with bounded reward ($\forall x(k), u(k) | R_k(x(k), u(k)) | \leq c$).³¹ The Q learning agent uses the training rule of the equation

$$Q_k(x(k), u(k)) \leftarrow R_k(x(k), u(k)) + \gamma \min_{u(k+1)} Q_{k+1}(x(k+1), u(k+1))$$

initializes its $Q_k(x(k), u(k))$ to arbitrary finite values, and uses a discount factor γ such that $0 \leq \gamma < 1$. Let $Q_k^{(n)}(x(k), u(k))$ denote the agent's hypothesis $Q_k(x(k), u(k))$ following the n th update. If each state-action pair is visited infinitely often, then $Q_k^{(n)}(x(k), u(k))$ converges to $Q_k(x(k), u(k))$ as $n \rightarrow \infty$, for all $x(k), u(k)$.

Remark 2. Lemma 1 provides a guarantee on convergence of reinforcement learning-based FTC. Using policy iteration which includes alternation procedures of policy evaluation and policy improvement, the Q learning agent will finally converge to the steady state and control $\pi^*(x(k), u(k))$ can be obtained readily.

Performance index compensator

The fault-tolerant reinforcement learning-based (RL) controller $\pi^*(x(k), u(k))$ has the capability of approaching to healthy performance index within its ranges. If the reinforcement learning-based controller (RLC) cannot achieve the goal of performance index, for instance, due to the actuator saturation, the compensator will enhance the FTC performance. The schematic block diagram with compensator is depicted in Figure 3.

The reference model works parallel to the plant unit and yields the healthy state variable $x_m(k)$ that is used to get the stage costs J_m under fault-free condition. The state variable $x_p(k)$ of the plant is used to get the stage costs J_p which include the fault and fault-free cases. The compensator is designed according to the error between J_p and J_m . The plant prevents its stage costs J_p from increasing by adjusting the set signal r to r_p via the compensator output r_c .

Denote by ΔJ the error between the stage costs J_p from the plant and the stage costs J_m from the reference model. One has

$$\begin{aligned} \Delta J &= J_p - J_m \\ &= \sum_{k=1}^{\infty} (x_p^T(k) M x_p(k) + r_p^T(k) N r_p(k)) \\ &\quad - \sum_{k=1}^{\infty} (x_m^T(k) M x_m(k) + r^T(k) N r(k)) \end{aligned} \quad (21)$$

From Figure 3, one can have

$$r_p = r - r_c \quad (22)$$

Substituting equation (22) into equation (21), one has

$$\begin{aligned} \Delta J &= \sum_{k=1}^{\infty} [x_p^T(k) M x_p(k) + r_p^T(k) N r_p(k) \\ &\quad - (x_m^T(k) M x_m(k) + (r_p(k) + r_c(k))^T N (r_p(k) + r_c(k)))] \\ &= \sum_{k=1}^{\infty} [(x_p^T(k) M x_p(k) - x_m^T(k) M x_m(k)) - r_c^T(k) N r_c(k)] \end{aligned} \quad (23)$$

Substituting $x_\Delta(k) = x_p(k) - x_m(k)$ into equation (23) yields

$$\Delta J = \sum_{k=1}^{\infty} (x_\Delta^T(k) M x_\Delta(k) - r_c^T(k) N r_c(k)) \quad (24)$$

For an online system, what we focus on is the error $\Delta J(k)$ of the performance index at every sampling time k

$$\Delta J(k) = x_{\Delta}^T(k) M x_{\Delta}(k) - r_c^T(k) N r_c(k) \quad (25)$$

Letting $\Delta J(k) = 0$, one has

$$r_c^T(k) N r_c(k) = x_{\Delta}^T(k) M x_{\Delta}(k) \quad (26)$$

There exists a nonsingular matrix $W \in \mathcal{R}^m$ such that

$$W^{-1} N W = \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \quad (27)$$

where Λ is a diagonal matrix and $\lambda_1, \lambda_2, \dots, \lambda_m$ are the eigenvalues of N .

Based on equations (26) and (27), one has

$$r_c^T(k) W \Lambda W^{-1} r_c(k) = x_{\Delta}^T(k) M x_{\Delta}(k) \quad (28)$$

Let $b = x_{\Delta}^T(k) M x_{\Delta}(k)$ and denote

$$y_1 = r_c^T(k) W \text{ and } y_2 = W^{-1} r_c(k) \quad (29)$$

where $y_1 = [y_{11}, y_{12}, \dots, y_{1m}]$ and $y_2 = [y_{21}, y_{22}, \dots, y_{2m}]^T$.

As a result, equations (28) and (29) imply

$$\lambda_1 y_{11} y_{21} + \lambda_2 y_{12} y_{22} + \dots + \lambda_m y_{1m} y_{2m} = b \quad (30)$$

Suppose that the first item is determined to compensate; therefore, $[y_{11}, y_{12}, \dots, y_{1m}]$ is chosen as a vector $[1, 0, \dots, 0]$ and $[y_{21}, y_{22}, \dots, y_{2m}]^T$ is computed as $[b / \lambda_1, *, \dots, *]^T$ according to equation (30).

From equation (30), we can have

$$r_c = W y_2 = W \left[\frac{b}{\lambda_1}, 0, \dots, 0 \right]^T \quad (31)$$

Remark 3

1. Equation (31) is a special solution of equation (26), which means $\Delta J = 0$.
2. A compensation r_c is determined by b , λ_1 , and W according to formula (31). Therefore, the compensation $r_c(k)$ is regulated according to $x_{\Delta}(k)$ for any given M and N .
3. The compensation aims to make the performance index of the real-time system under faulty conditions track the performance index of the reference model.
4. In the fault-free case, $x_p(k) = x_m(k)$ if the reference model is accurate enough to ignore an error. Therefore, $x_{\Delta}(k) = x_p(k) - x_m(k) = 0$. Furthermore, $b = x_{\Delta}^T(k) M x_{\Delta}(k) = 0$ and $r_c = 0$ according to formula (31). This means that it does not need any

compensation. Summarily, the compensator is equipped online for both scenarios: fault or fault free.

The reinforcement learning-based fault tolerant control (RL-FTC) algorithm can be summarized as follows:

Step 1. Calculate $x_{\Delta}(k)$ according to $x_{\Delta}(k) = x_p(k) - x_m(k)$;

Step 2. Initialize $Q(x_{\Delta}(k), u(k))$ to zero;

Step 3. Select an action $u(k)$ randomly;

Step 4. Compute the compensation r_c according to formula (31);

Step 5. Receive immediate reward $R(x_{\Delta}(k), u(k))$ according to

$$R = x_{\Delta}^T(k) M x_{\Delta}(k) + (r(k) - r_c(k))^T N (r(k) - r_c(k));$$

Step 6. Observe the new state $x_{\Delta}(k+1)$ and update $Q(x_{\Delta}(k+1), u(k+1))$ based on the current state $x_{\Delta}(k)$ according to formula (18);

Step 7. Set the next state $x_{\Delta}(k+1)$ as the current state $x_{\Delta}(k)$;

Step 8. Find the best action π^* according to the formula (19);

Step 9. Repeat Steps 5–8 until it is convergent.

Further analysis on compensation

Actually, the proposed FTC method does not need any detailed information on the system dynamics. However, for the purpose of further analysis, we adopt the model expression to the theoretical analysis and assume that the time series of the plant can be described by the following discrete time form

$$x_p(k+1) = A x_p(k) + B r_p(k) + \omega(k, x_p, r_p) + f(k) + u(k) \quad (32)$$

where $x_p(k) \in \mathcal{R}^n$ is the state vector, $r_p(k) \in \mathcal{R}^m$ is the control input vector, A and B are the parameter matrices with appropriate dimensions, $f(k)$ represents a fault, $u(k)$ is the control signal produced by reinforcement learning, and $\omega(k, x_p, r_p) \in \mathcal{R}^n$ is a real nonlinear vector function with Lipschitz constants α_1 and α_2 , that is

$$\|\omega(k, \hat{x}, \hat{r}) - \omega(k, x, r)\| \leq \alpha_1 \|\hat{x} - x\| + \alpha_2 \|\hat{r} - r\| \quad (33)$$

$$\forall (k, \check{x}, r), (k, x, r) \in \mathcal{R} \times \mathcal{R}^n \times \mathcal{R}^m$$

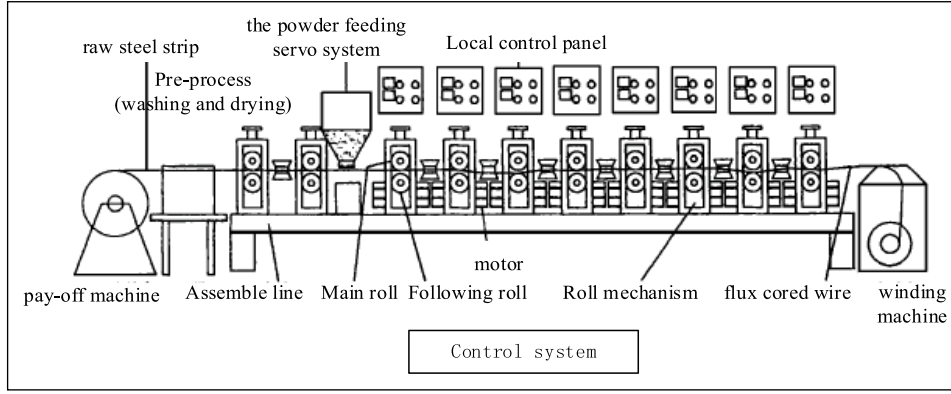


Figure 4. The system scheme of flux cored wire.

It is noticed that $x_{\Delta}(k) = x_p(k) - x_m(k)$ and $r(k) = r_p(k) + r_c(k)$. Therefore, equation (26) can be rewritten as

$$\begin{aligned} x_m(k+1) &= Ax_m(k) + B[r(k) - r_c(k)] \\ &\quad - x_{\Delta}(k+1) + Ax_{\Delta}(k) + f(k) \\ &\quad + u(k) + \omega(k, x_p, r_p) \end{aligned} \quad (34)$$

The reference model dynamics can be extracted as follows

$$x_m(k+1) = Ax_m(k) + Br(k) + \omega(k, x_m, r_m) \quad (35)$$

Therefore, from equation (34), we can have

$$\begin{aligned} x_{\Delta}(k+1) &= Ax_{\Delta}(k) - Br_c(k) + f(k) + u(k) \\ &\quad + \omega(k, x_p, r_p) - \omega(k, x_m, r_m) \end{aligned} \quad (36)$$

Subtracting $x_{\Delta}(k)$ from both sides of equation (36), one has

$$\begin{aligned} x_{\Delta}(k+1) - x_{\Delta}(k) &= (A-I)x_{\Delta}(k) - Br_c(k) \\ &\quad + f(k) + u(k) + \omega(k, x_p, r_p) \\ &\quad - \omega(k, x_m, r_m) \end{aligned} \quad (37)$$

If

$$\begin{aligned} (A-I)x_{\Delta}(k) - Br_c(k) + \omega(k, x_p, r_p) \\ - \omega(k, x_m, r_m) + f(k) + u(k) = 0 \end{aligned} \quad (38)$$

then one has

$$\begin{aligned} (A-I)x_{\Delta}(k) - Br_c(k) + \omega(k, x_p, r_p) \\ - \omega(k, x_m, r_m) = -u(k) - f(k) \end{aligned} \quad (39)$$

Furthermore, one has

$$\begin{aligned} \|-u(k) - f(k)\| &= \|(A-I)x_{\Delta}(k) - Br_c(k) \\ &\quad + \omega(k, x_p, r_p) - \omega(k, x_m, r_m)\| \\ &\leq (\|A-I\| + \alpha_1)\|x_{\Delta}(k)\| \\ &\quad + (\|B\| + \alpha_2)\|r_c(k)\| \end{aligned} \quad (40)$$

which is equivalent to

$$\begin{aligned} (\|A-I\| + \alpha_1)\|x_{\Delta}(k)\| &\geq \|-u(k) - f(k)\| \\ &\quad - (\|B\| + \alpha_2)\|r_c(k)\| \end{aligned} \quad (41)$$

As a result, there is $x_{\Delta}(k) = 0$ if

$$\|-u(k) - f(k)\| \leq (\|B\| + \alpha_2)\|r_c(k)\| \quad (42)$$

Clearly, $x_{\Delta}(k) = 0$ indicates the real-time plant after the compensation/control should be equivalent to the reference model. The reference model is assumed to be stable, and therefore the real-time plant under control/compensation is also stable.

Remark 4. The inequity (equation (42)) indicates that a controller $u(k)$ and a compensator $r_c(k)$ should be found to mitigate the adverse influences from the fault $f(k)$. When without compensation, that is, $r_c(k) = 0$, formula (42) becomes

$$\|-u(k) - f(k)\| \leq 0 \quad (43)$$

When comparing equation (42) with equation (43), it is clear that $u(k)$ in equation (43) can be easier to mitigate the fault with the aid of the compensator: $r_c(k) \neq 0$. It indicates that a system with compensation has more fault-tolerant ability than that without compensation.



Figure 5. Test bed of flux cored wire.

Experimental results

Flux cored wire, also called tubular welding wire, is an important welding material that is used for different applications by adjusting alloy composition and type. The process scheme is shown in Figure 4. First, the raw steel strip of pay-off machine is preprocessed by washing and drying. After that, the post-processed steel strips that are mixed up with the given powder using the powder feeding servo system begin to roll. After several rolling stages, the flux cored wires are produced. Finally, they are synchronously collected as the product by a wire winding machine. All the processes are controlled by the computer control system. There are some restrictions in the production process for flux cored wire such as the maximum tension of 0.4 mm/m and the error of powder content should be equal to or less than 1%. These performance criteria are realized by quality speed control. False or mistakes may cause wire break or quality rejection.

We focus on the rolling system because it is the core of the system. Moreover, we also abandon the preprocessing and the powder feeding servo system because they have little influence on the rolling system. The rolling system consists of a driven subsystem by several gear motors and an oppression subsystem that has an independent control system which provides proper pressing force according to tension detection. The oppression subsystem is considered as a disturbance to the driven subsystem for the target of speed control in the flux cored wire process. Therefore, the oppression subsystem is not considered in our test. The test bed of flux cored wire shown in Figure 5 is driven by four AC motors (M1, M2, M3, and M4) with a MultiQ PCI Data Acquisition board, a control board, and a data acquisition and control board (DACB) interface board from Quanser. The character of test bed depends on the AC motors. Every single motor has its independent control channel in order to imitate a split drive-type system. All motors work well by the proportional–integral–derivative (PID) speed output feedback in the healthy state and the rollers also operate well.

The motor M4 is selected to test our approach under the condition that the motors M1–M3 operate normally. The motor M4 within a range of 0–2000 r/min is driven by a frequency transformer with a control input of voltage. The computer provides a controllable speed range of ± 250 r/min

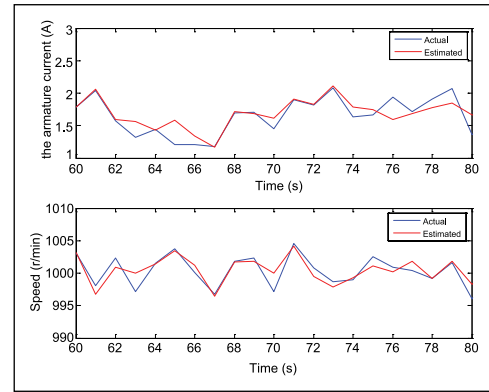


Figure 6. The estimated states by FNN and the actual state data (from 60 to 80).

which converts to the voltage signal of -5 to 5 V by an analog conversion channel from Quanser. The target of the driven subsystem is to keep the roller rotational speed n_{roll} following a reference speed n_r that is specified by the craft with considerations of prescription, filling rate, materials, and productivity. It is manually input through the computer as the reference input of the system. The state variables of the system are selected as the motor speed n_a and the armature current i_a which are measurable and collected by sensors. The output variable of the system is the roller rotational speed n_{roll}

$$n_{roll} = K_{gear} n_a \quad (44)$$

where K_{gear} is the transmission ratio and $K_{gear} = 12$ in the test bed. There is no sensor installed to measure n_{roll} because it keeps pace with the motor speed n_a and is also easy to be obtained according to equation (44). The FTC is to recover the rotational speed of the roller to the reference speed in the condition of fault, that is

$$\Delta J = n_{roll} - n_r \rightarrow 0$$

where ΔJ is the performance index that indicates the difference of the roller speed between n_{roll} and n_r .

For a flux cored wire system, the reference speed n_r is fixed. Once the reference speed n_r is determined, the control system will make a contribution to the system with little fluctuation in the fault-free condition. These fluctuations depend mainly on the load variation and the tracing delay of the frequency transformer that are considered as the inner disturbances. As a result, the states have enough information to forecast the next state for the driven subsystem even under the function of the controller. In this condition, the objection and the controller can be regarded as an integer system and their interactions are considered as inner events. As a result, the motor speed n_a and the armature current i_a are chosen as the inputs and outputs of the NN at the previous sampling k and at the next sampling $k+1$, respectively. The output variable n_{roll} is abandoned because it is proportional to the motor

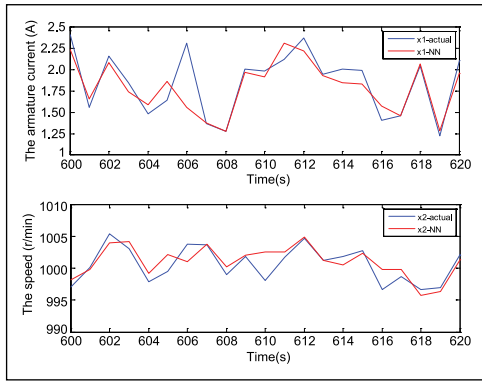


Figure 7. The estimated states by FNN and the actual state data (from 600 to 620).

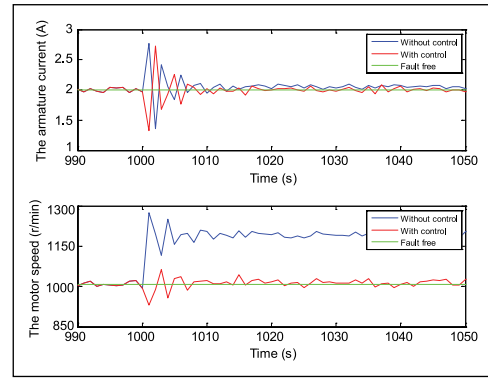


Figure 10. State evolution.

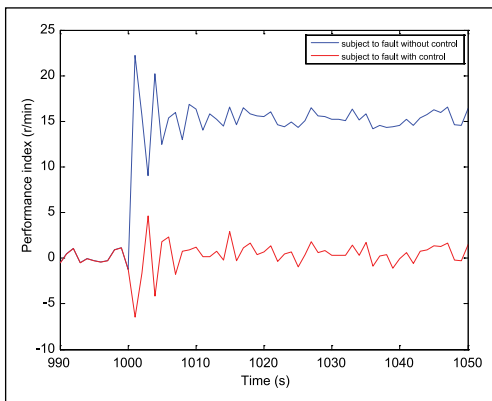


Figure 8. The error ΔJ of the performance indices.

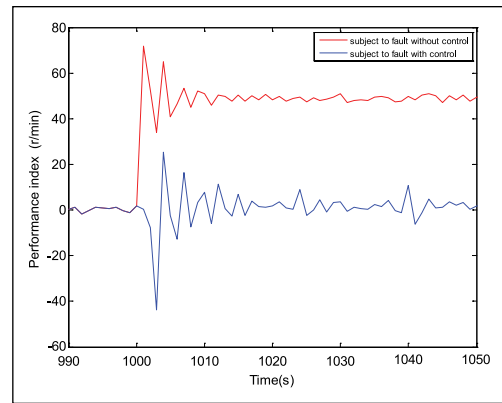


Figure 11. The error ΔJ of the performance indices.

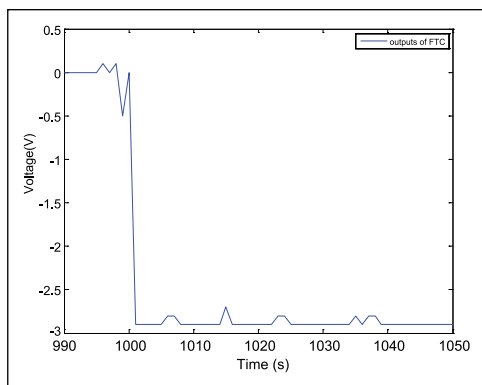


Figure 9. The output of RL controller.

speed n_a and has no measure. A reference model is described in the form of a feedforward neural network (FNN) with 2-10-2 structure. There are a sample set of 1000 data collected in the fault-free case to apply to train. The FNN is trained offline by Levenberg–Marquardt algorithm.³³

A period from 60 to 80 is selected randomly to test the effect of FNN. The estimated states by FNN and the original healthy data are shown in Figure 6. The blue line represents the original data and the red line represents the output of NN. The estimated states by FNN is consistent with the original data.

The estimated states by FNN and the original healthy data of another period from 600 to 620 are shown in Figure 7. One can see that the two curves show good consistency. Therefore, the well-trained FNN can be used as a black box reference model.

FTC: sensor fault scenarios

A speed step fault with an amplitude of 180 r/min. A step fault of speed sensor with an amplitude of 180 r/min that is within the controller’s regulations added to the control system with the sampling number of 1000. The controller is designed by reinforcement learning and its output is $|u| \leq 5V$ with the consideration of saturation. The step of episode is chosen as 10 and γ as 0.9.

Figure 8 shows the evolution of error ΔJ of the performance indices. The curves with and without control coincide before a fault occurs. Without tolerant control, one can see that the performance index has a significant difference after the fault occurs. Conversely, with tolerant control, the performance indices derive small errors after the fault occurs. The RL learning control is shown in Figure 9. Compared with the scenario without the RL control, the performance index has been improved much. The system states are shown in Figure 10. It is seen from Figure 10 that FTC can mitigate the adverse influences from the faults, which can be seen more clearly for the motor speed.

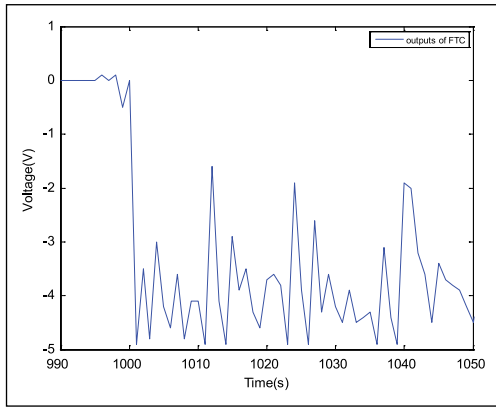


Figure 12. The output of RL controller.

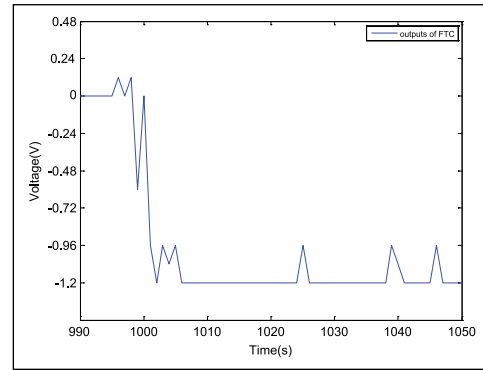


Figure 15. The output of RL controller.

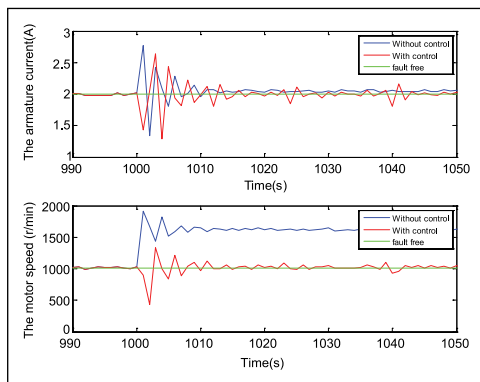


Figure 13. State evolution.

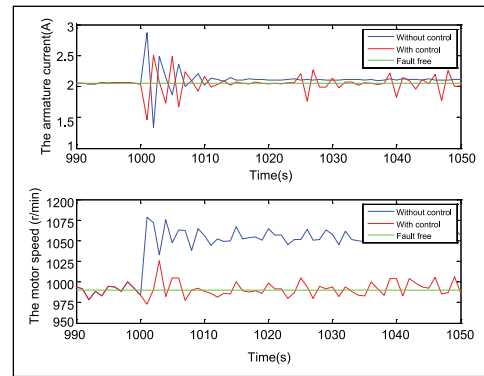


Figure 16. State evolution.

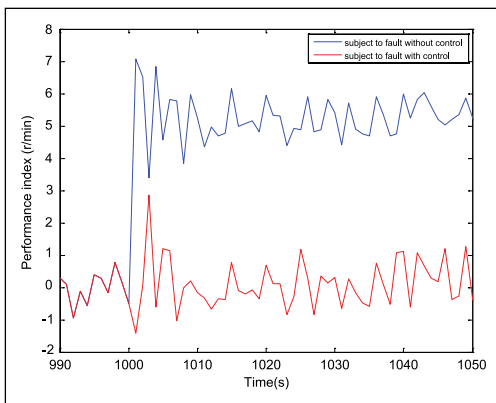


Figure 14. The error ΔJ of the performance indices.

A speed step fault with an amplitude of 600 r/min. The amplitude of the fault is enlarged to 600 r/min which means that it is out of the range of the controller, that is, 250 r/min. The error ΔJ of the performance indices, the output of RLC, and the state evolution are shown in Figures 11–13, respectively. From Figure 11, one can see that the steady error ΔJ values of the performance indices are, respectively, 50 (red curve: without control) and almost 0 r/min with fluctuation (blue curve: with control). From Figure 13, one can see that tolerant control can make the motor speed recover from 1600 down to 1000 r/min after the fault occurs.

FTC: actuator fault scenario

An error of speed actuator with an amplitude of 60 r/min is also tested (the control signal has the voltage of 1.2 V). Figure 14 shows the evolution of error ΔJ of the performance indices, implying a better performance index with RL control than without control. The output of RL controller is shown in Figure 15. The state evolution is depicted in Figure 16. With the function of RL learning control, the motor speed is keeping the trace fault free (red line and green line) compared to the motor speed without control (blue line) when a speed actuator fault occurs. The armature current without any load changes is also keeping stable except suffering a transient process caused by the disturbance of actuator false.

Conclusion

The reinforcement learning is a data-driven online method which directs to the goal by iterating value evaluation and policy improvement, which can achieve an optimal action without knowing any system dynamic characteristics that are difficult to know at the beginning when a fault occurs. In this paper, we have shown that reinforcement learning is an effective approach to solve the FTC problem when the faulty information is unavailable to the designers. The proposed reinforcement learning-based FTC controller has been applied to a flux cored wire system, and the effectiveness has been well demonstrated. It is worthy to point out

that the proposed FTC is real-time RL learning with the aid of the reference model to track the performance index without any fault. The reference model was identified by FNN instead of difficult mechanism modeling. As a result, the method used is data driven in essence. If an explicit reference model cannot be obtained for a complicated system, an implicit model could be obtained by data-driven learning which can be used as the reference. It would be of interest to develop a reference model-free data-driven FTC algorithm at the cost of the system performance in the future.


Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors would like to acknowledge the research support from China Scholarship Council, the Alexander von Humboldt Renewed Stay Fellowship, and the E&E faculty at the University of Northumbria.

ORCID iD

Dapeng Zhang  <https://orcid.org/0000-0002-9657-0088>

References

- Blanke M, Izadi-Zamanabadi R, Bogh SA, et al. Fault-tolerant control systems: a holistic view. *Control Eng Pract* 1997; 5(5): 693–702.
- Gao Z, Cecati C and Ding SX. A survey of fault diagnosis and fault-tolerant techniques—part I: fault diagnosis with model-based and signal-based approaches. *IEEE T Ind Electron* 2015; 62(6): 3757–3767.
- Yu X and Jiang J. A survey of fault-tolerant controllers based on safety-related issues. *Annu Rev Control* 2015; 39: 46–57.
- Zhang Y and Jiang J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annu Rev Control* 2008; 32(2): 229–252.
- Yin S, Xiao B, Ding SX, et al. A review on recent development of spacecraft attitude fault tolerant control system. *IEEE T Ind Electron* 2016; 63(5): 3311–3320.
- Jiang J and Yu X. Fault-tolerant control systems: a comparative study between active and passive approaches. *Annu Rev Control* 2012; 36(1): 60–72.
- Gao Z, Breikin T and Wang H. Reliable observer-based control against sensor failures for systems with time delays in both state and input. *IEEE T Syst Man Cy A* 2008; 38(5): 1018–1029.
- Van M, Ge SS and Ren H. Robust fault-tolerant control for a class of second-order nonlinear systems using an adaptive third-order sliding mode control. *IEEE T Syst Man Cy: S* 2017; 47(2): 221–228.
- Li Y and Yang G. Adaptive fuzzy decentralized control for a class of large-scale nonlinear systems with actuator faults and unknown dead zones. *IEEE T Syst Man Cy: S* 2017; 47(5): 729–740.
- Zhao Z, Yang Y, Ding SX, et al. Fault-tolerant control for systems with model uncertainty and multiplicative faults. *IEEE T Syst Man Cy: S*. Epub ahead of print 19 October 2017. DOI: 10.1109/TSMC.2017.2759144.
- Ding SX, Wang Y, Yin S, et al. Data-driven design of fault-tolerant control systems. *IFAC P Vol* 2012; 45: 1323–1328.
- Yin S, Luo H and Ding SX. Real-time implementation of fault-tolerant control systems with performance optimization. *IEEE T Ind Electron* 2014; 61(5): 2402–2411.
- Yin S, Gao H, Qiu J, et al. Adaptive fault-tolerant control for nonlinear system with unknown control directions based on fuzzy approximation. *IEEE T Syst Man Cy: S* 2017; 47(8): 1941–1952.
- Wang Z, Liu L and Zhang H. Neural network-based model-free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault. *IEEE T Syst Man Cy: S* 2017; 47(8): 2351–2362.
- Yang J, Sun Z and Chen Y. Fault detection using the clustering-kNN rule for gas sensor arrays. *Sensors* 2016; 13(12): 2069.
- Gao X and Hou J. An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process. *Neurocomputing* 2016; 174: 906–911.
- Santos P, Villa LF, Renones A, et al. An SVM-based solution for fault detection in wind turbines. *Sensors* 2015; 15(3): 5627–5648.
- Yao Y and Gao F. A survey on multistage/multiphase statistical modeling methods for batch processes. *Annu Rev Control* 2009; 33(2): 172–183.
- Zhu D, Bai J and Yang SX. A multi-fault diagnosis method for sensor systems based on principle component analysis. *Sensors* 2010; 10(1): 241–253.
- Lei Y, He Z and Lin J. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mech Syst Signal Pr* 2013; 35(1–2): 108–126.
- Feng Z, Liang M and Zhang Y. Fault diagnosis for wind turbine planetary gearboxes via demodulation analysis based on ensemble empirical mode decomposition and energy separation. *Renew Energ* 2012; 47: 112–126.
- Macgregor J and Cinar A. Monitoring, fault diagnosis, fault-tolerant control and optimization: data driven methods. *Comput Chem Eng* 2012; 47: 111–120.
- Xie C and Yang G. Data-based fault-tolerant control for uncertain linear systems with actuator faults. *IET Control Theory A* 2016; 10(3): 265–272.
- Wang J and Yang G. Data-driven output-feedback fault-tolerant L2 control of unknown dynamic systems. *ISA T* 2016; 63: 182–195.
- Wang Z, Liu L, Zhang H, et al. Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm. *IEEE T Syst Man Cy: S* 2016; 46(5): 611–622.
- Kaelbling LK, Littman ML and Moore AW. Reinforcement learning: a survey. *J Artif Intell Res* 1996; 4: 237–285.
- Lewis FL, Vrabie D and Vamvoudakis KG. Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. *IEEE Contr Syst Mag* 2012; 32(6): 76–105.
- Vamvoudakis KG. Q-learning for continuous-time linear systems: a model-free infinite horizon optimal control approach. *Syst Control Lett* 2017; 100: 14–20.
- Bhatnagar S and Babu K. New algorithms of the Q-learning type. *Automatica* 2008; 44(4): 1111–1119.
- Watkins JCH and Dayan P. Q-learning. *Mach Learn* 1992; 8: 279–292.
- Sutton R and Barto A. *Reinforcement learning: an introduction*. Cambridge, MA: The MIT Press, 2005.
- Bradtke S and Ydstie BE. Adaptive linear quadratic control using policy iteration. *P Amer Contr Conf* 1994; 3: 3475–3479.
- Ngia LSH and Sjöberg J. Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm. *IEEE T Signal Proces* 2000; 48(7): 1915–1927.