A Discrete Model-Free Scheme for Fault Tolerant Tracking Control of Redundant Manipulators

Ning Tan, *Senior Member, IEEE*, Zhaohui Zhong, Peng Yu, Zhan Li, and Fenglei Ni *Member, IEEE*

***Abstract*—Fault tolerance is a critical requirement for robust motion control of redundant robotic manipulators. This task aims to endow the redundant manipulator with the capability to achieve the required path of end-effector in the condition that one or some of its joints' motion fail. Although many fault-tolerant control algorithms of redundant manipulator have been proposed in recent years. However, few of them are on the basis of the condition that the robotic model is unknown. The complexity of the calculation model limits the efficiency and portability of these algorithms. For the first time, we proposed a discrete model-free fault tolerant tracking control (DMFFTC) scheme of redundant manipulator, which takes into account the fault tolerance in the control system of redundant manipulator by formulating it into a quadratic programming (QP) framework. The core of the proposed scheme consists of a discrete kinematic estimator and a discrete QP solver, powered by which the fault-tolerant control problem is transformed into a unified computing problem relaxing the need of knowing the redundant manipulator's kinematic model. A discrete joint space observer is proposed for detection of the happening of faulty states. Extensive simulations and experiments based on a redundant manipulator are performed and analyzed to support the verification of the efficiency and effectiveness of the proposed scheme.**

***Index Terms*—Fault tolerance, model-free, zeroing neural network, tracking control, redundant manipulator.**

I. Introduction

REDUNDANT manipulators are playing an important part in modern manufacturing nowadays. They perfectly replace humans in some repetitive, laborious, and dangerous tasks, such as welding, ironing and painting [1]. Except for accomplishing versatile tasks, the redundant manipulator possesses additional capabilities such as fault tolerance [2], obstacle avoidance [3], joint-limit avoidance [4]. That is because the degrees of freedom (DOF) of the redundant manipulator are more than the required ones when performing the given task. Thus redundant manipulator have higher flexibility than non-redundant manipulators. It is also common that the fault of a certain or even a few joints of the redundant manipulator occurs. On the one hand, if we force the fault joint to continue working, the manipulator may fail to complete the prescribed path-tracking task, and it may cause serious injury [5]. On the other hand, repair or replacement is expensive, unnecessary or even impossible for redundant manipulators with this problem. Because the remaining intact joints of the manipulator may still be able to complete the existing tasks well. It makes sense that if the desired control task can be accomplished well by only adjusting some parameters instead of reconsidering a new control strategy. Thus, it is very important to plan the redundant manipulators to complete the tasks even in the condition that some joints fault (or are locked) to function properly. The set of techniques to achieve this function is called fault tolerant control.

So far, a lot of fault-tolerant control techniques of redundant manipulator have been developed. In general, the existing fault-tolerance methods can be divided into passive fault-tolerant methods and active fault-tolerant methods [6]. The passive method designs and fixes the corresponding controllers to make it robust to fault, and does not require fault detection or controller reconfiguration. However, this method has limited capabilities for fault-tolerant control. The active method actively responds to system failures by constantly reconfiguring control actions to maintain the stability of the entire system [7]. However, most of the active methods assume that the robot manipulator's model are known *a priori*.

The correctness and accuracy of the solution obtained by these methods will be directly affected by the correctness and accuracy of the models which are equivalent to the Jacobian matrices in kinematic cases. Hence, corresponding parameters in the fault-tolerant algorithm need to be modified according to different manipulators used. In addition, the Jacobian matrix derived from the forward kinematics model might be different from the actual Jacobian matrix due to calculation errors. Moreover, for robotic manipulators with elastic elements (e.g., those composed of active rigid joints and deformable links [8]) or flexible-bodied manipulators (e.g., soft and continuum robots [9], [10]), calculating their accurate models would be extremely complicated and inaccurate. Even for the traditional rigid-link robotic manipulators, obtaining an exact model of the system is often laborious and time-consuming through a calibration and parameter identification process, or even impossible in practical processes [11], [12]. This may cause issues in the application and deteriorate the accuracy of the solution. Thus, algorithms mentioned above

This work is partially supported by the State Key Laboratory of Robotics and Systems (HIT) (SKLRS-2021-KF-07), the National Natural Science Foundation of China (62173352, U181126), the Guangdong Basic and Applied Basic Research Foundation (2021A1515012314), the Research Fund of Guangdong Key Laboratory of Precision Equipment and Manufacturing Technique (PEMT202104), the Open Project of Shenzhen Institute of Artificial Intelligence and Robotics for Society (AC01202005006), and the Key-Area Research and Development Program of Guangzhou (202007030004). (*Corresponding authors: Ning Tan; Fenglei Ni*)

N. Tan is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, PR China and State Key Laboratory of Robotics and Systems (HIT), Harbin, PR China. (e-mail: tann5@mail.sysu.edu.cn)

Z. Zhong and P. Yu are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, PR China.

Z. Li is with the Department of Computer Science, Swansea University, Swansea, U.K.

F. Ni is with the State Key Laboratory of Robotics and Systems (HIT), Harbin, PR China. (e-mail: flni@hit.edu.cn)

are not well transplanted to various redundant manipulators. The joints of the manipulator may also degenerate and cannot perfectly execute the input instructions. However, the model-free feature allows manipulator to ignore the changes in the kinematic model caused by the degeneration of the joints, so that the manipulator can still complete the task.

In view of the motivation and inspiration mentioned above, here we propose a discrete model-free fault tolerant tracking control (DMFFTC) scheme of redundant manipulator. We simulated the situation of loosing one or two joints's actuation of the manipulator. Experiments are also carried out to verify our scheme using a lightweight redundant manipulator called ”Kinova Jaco2”. It needs to be emphasized that the proposed scheme is based on zeroing neural networks, so its computing power can be really used after making hardware circuits. The main contributions of this paper are summarized as follows.

- For the first time, the model-free fault tolerant control of redundant manipulators is proposed through formulating the fault tolerant control as a QP problem. This QP problem is transformed into a linear variational inequality which can be solved by a projection-operator-based neural network (POBNN).
- Different from the state-of-the-art methods, a data-driven kinematic-model estimator is designed and integrated into the fault tolerant control scheme. Based on zeroing neural networks, a neurodynamic system is developed to approximate the the Jacobian matrix of the redundant manipulator.
- Furthermore, two main equations of the proposed scheme are optimized discretely and two discrete solvers are proposed. The theoretical analysis of the two solvers are provided. By combining the two discrete solvers, a DMFFTC scheme is proposed.
- A discrete joint space observer is proposed for detection of the happening of faulty states.
- Simulations and experiments with locking one joint and two joints are presented respectively to demonstrate the efficiency and superior performance of the DMFFTC scheme.

The rest of the paper is organized as follows: we firstly review other related work on fault tolerant control of robot manipulators in Section II. The continuous QP scheme and the discrete QP solver are presented in Section III. Section IV presents the discrete kinematic estimator used in the fault tolerant scheme. A discrete joint state observer is presented in Section V. Section VI and VII give simulation and experiment results on the performance verification of the DMFFTC scheme. Finally we make a conclusion on the achievement and point out future direction in Section VIII.

II. Related Work

Due to the difficulty to find and generalize an analytical solution at the joint level, the current methods directly deal with the strong coupling nonlinear problems in the kinematics modeling by using the pseudoinverse of the Jacobian matrix of the manipulator [13], [14]. In order to leverage the improved computing power of real-time implementation to motion control and to realize the secondary performance indicators, many methods are proposed to achieve multifarious goals. For example, joint motion limits, rectification joint angular drift, and limit mechanical energy consumption. On the basis of ensuring accurate motion control, the effects of the parallel processing method based on constraint optimization on different performance indexes are studied [4]. Chen and Zhang [15] proposed a robust zeroing neural-dynamics approach as well as its associated model for solving the inverse kinematics problem of mobile robot manipulators. Jin *et al.* [16] designed a time-varying QP with inherent tolerance to noises scheme. Guo and Zhang [17] developed and investigated a weighted velocity and acceleration minimization scheme to prevent the occurrence of high joint velocity and joint acceleration in redundant robot manipulators. With help of these techniques, inverse kinematics problems of redundant manipulators have been successfully and efficiently resolved, which give us a good foundation for the study of fault-tolerant control.

Most of works on fault-tolerant control of robot manipulators are based on pseudoinverse-type methods [5], [18], [19] as well. However, pseudoinverse-type methods cannot efficiently overcome Jacobian-matrix singularities when the manipulator joint fail [18]. Li and Zhang [2] considered a fault-tolerant constraint in the optimization criteria for motion planning for redundant manipulators. In order to facilitate the solution, this fault-tolerant constraint is transformed to a QP problem. Zhang *et al.* [20] also proposed a fault-tolerant scheme for redundant manipulator based on QP problem. The main focus of their work is on solving the problem at the acceleration level. In order to reduce initial position error of the end-effector, Li *et al.* [21] developed a fault-tolerant motion planning scheme which can diminish the initial position error arising in the manipulator state adjustment. Li *et al.* [22] designed and implemented a fault-tolerant motion planning scheme which can adaptively localize which joints run away from the normal state to be fault. A recurrent neural network (VP-RNN) with varying parameters was developed to resolve the fault-tolerant motion planning problem, which can make the remaining healthy joints to remedy the whole system which is effected by faulty joints and complete the expected end-effector path [23]. By taking both the robot kinematics and robot dynamics into account, a different-level simultaneous minimization scheme is proposed to solve fault-tolerant motion planning of redundant manipulator using a recurrent neural network [24]. However, all these schemes deal with fault-tolerant problem of redundant manipulators in the condition that manipulators' models are known *a priori*. As far as authors' knowledge, no related work is available on model-free fault tolerant tracking control problem of redundant manipulators.

In response to the challenge of unknown or uncertain kinematic models, there have been numerous works proposed in the past decades. For examples, Cheah *et al.* proposed an approximate Jacobian adaptive controller for trajectory tracking of robot with uncertain kinematics, dynamics and actuator model [25]. Ahmadipour *et al.* provided an adaptive task-space controller for rigid-link robots with uncertain kinematics and dynamics [26]. Xiao *et al.* established a theoretical framework for designing an observer-based controller to achieve exponen-

notation	description
$\boldsymbol{\theta}$	joint vector of manipulator
$\dot{\boldsymbol{\theta}}$	joint angular velocity of manipulator
$\ddot{\boldsymbol{\theta}}$	joint acceleration of manipulator
$\boldsymbol{\theta}_r$	joint vector of redundant manipulator measured by sensors
$\boldsymbol{r}_e$	end-effector's position and orientation vector
$\dot{\boldsymbol{r}}_e$	velocity of the end-effector
$\ddot{\boldsymbol{r}}_e$	acceleration of the end-effector
$\boldsymbol{r}_d$	desired path
$\dot{\boldsymbol{r}}_d$	velocity of the desired path
$\boldsymbol{J}$	Jacobian matrix of the manipulator
$\boldsymbol{A}$	joint condition matrix used for fault tolerance
$\boldsymbol{\xi}^-$	lower physical joint limits
$\boldsymbol{\xi}^+$	upper physical joint limits
$\boldsymbol{y}$	dual variable of $\dot{\theta}$
$\boldsymbol{y}_r$	dual variable of $\dot{\theta}_r$
$\hat{\boldsymbol{y}}_r$	estimate variable of $\boldsymbol{y}_r$
$\tilde{\boldsymbol{y}}_r$	distance variable of $\boldsymbol{y}_r$
ϕ	sampling gap
$\boldsymbol{\Lambda}$	actual joint condition matrix used for observer
$\hat{\boldsymbol{\Lambda}}$	estimate variable of $\boldsymbol{\Lambda}$
$\tilde{\boldsymbol{\Lambda}}$	distance variable of $\boldsymbol{\Lambda}$

tial tracking control of robotic manipulators subject to both uncertain dynamics and uncertain kinematics [27]. Yang *et al*. developed an adaptive fuzzy control scheme for a dual-arm robot, where an approximate Jacobian matrix is applied to address the uncertain kinematic control [28]. Furthermore, Yang *et al*. developed a control scheme for interaction between robot manipulators and human arms in physical collision [29] where, to deal with the tracking problem of the uncertain manipulator, an adaptive controller was designed combining with the radial basis function neural network. Zhang *et al*. proposed a method capable of simultaneously optimizing performance indices subject to physical constraints and handling parameter uncertainty [30]. However, this work requires the structure of the model to be known *a priori* with only unknown parameters. Overall, these precedents do not jumped out of limitations of the manipulators' model, nor do they support fault-tolerant control. A control scheme capable of performing high quality of tracking with fault tolerant capability in a model-free fashion is in urgently needed.

III. Fault Tolerance Scheme

A. QP Formulation for Fault Tolerance

For a redundant manipulator, the kinematics equation mapping the joint vector $\boldsymbol{\theta}(t) \in \mathbb{R}^n$ and the end-effector's position and orientation vector $\boldsymbol{r}_e(t) \in \mathbb{R}^m$ in Cartesian space can be written as below:

$$f(\boldsymbol{\theta}(t)) = \boldsymbol{r}_e(t) \tag{1}$$

where $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ in this formula represents a nonlinear mapping. Given $\boldsymbol{r}_e(t)$, it is usually very difficult to obtain the $\boldsymbol{\theta}(t) \in \mathbb{R}^n$ due to the nonlinearity and redundancy of $f(\cdot)$. Thus, the inverse-kinematic problem of redundant manipulator is usually solved at velocity level. Through differentiating both sides of (1) with respect to t, we can get the relationship between the change rate of actuators and the velocity of end-effector:

$$\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{r}}_e \tag{2}$$

where $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{n*m} = \partial f(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ is the Jacobian matrix of the redundant manipulator. To keep the paper concise, we use $\boldsymbol{J}$ instead of $\boldsymbol{J}(\boldsymbol{\theta})$ below. $\dot{\boldsymbol{\theta}}$ is the derivative of $\boldsymbol{\theta}$, which represents the joint angular velocity of the redundant manipulator. $\dot{\boldsymbol{r}}_e$ is the derivative of $\boldsymbol{r}_e$, which represents the velocity of the end-effector at time t along the actual path. Our goal is to control the redundant manipulator to move along the desired path $\boldsymbol{r}_d$. As introduced in [31], to solve this problem, a QP method (QP1) is used to solve the inverse kinematics problem of the manipulator:

$$\text{minimize } \boldsymbol{\Phi}(t) = \frac{1}{2}\dot{\boldsymbol{\theta}}^T \boldsymbol{W} \dot{\boldsymbol{\theta}} + \boldsymbol{q}^T \dot{\boldsymbol{\theta}} \tag{3}$$

$$\text{subject to } \boldsymbol{J}\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{r}}_d \tag{4}$$

where superscript T denotes the transpose of a matrix or vector; $\dot{\boldsymbol{r}}_d$ is the derivative of $\boldsymbol{r}_d$. $\boldsymbol{W}$ and $\boldsymbol{q}$ are defined specifically according to different control schemes. In this paper, as the fault-tolerant performance is mainly concerned, the coefficient matrix $\boldsymbol{W}$ is simply set as identity matrix $\boldsymbol{I}$, and coefficient vector $\boldsymbol{q} = 0$ [32].

As we introduced before, without fault-tolerant control, joint failures may cause serious consequences. Therefore, fault tolerant need to be considered in QP1. If there are m_f joints faulty (e.g., the ith, $\cdots$, and jth joints, $i, j \in [1, 2, \cdots, n]$), the corresponding joint-velocity variables should be forced set to zeros (e.g., $\dot{\boldsymbol{\theta}}_i = 0, \cdots, \dot{\boldsymbol{\theta}}_j = 0$). Thus, the following equations need to be guaranteed:

$$\begin{gathered} 0\dot{\boldsymbol{\theta}}_1 + \cdots + 1\dot{\boldsymbol{\theta}}_i + \cdots + 0\dot{\boldsymbol{\theta}}_{n-1} + 0\dot{\boldsymbol{\theta}}_n = 0, \\ \vdots \\ 0\dot{\boldsymbol{\theta}}_1 + 0\dot{\boldsymbol{\theta}}_2 + \cdots + 1\dot{\boldsymbol{\theta}}_j + \cdots + 0\dot{\boldsymbol{\theta}}_n = 0. \end{gathered} \tag{5}$$

The above equations can be rewritten as matrix form below:

$$\boldsymbol{A}\dot{\boldsymbol{\theta}} = 0, \tag{6}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ & & & \vdots & & & \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m_f \times n}, \tag{7}$$

$$\dot{\boldsymbol{\theta}} = \left[\dot{\boldsymbol{\theta}}_1, \cdots, \dot{\boldsymbol{\theta}}_i, \cdots, \dot{\boldsymbol{\theta}}_j, \cdots, \dot{\boldsymbol{\theta}}_n\right]^T = 0. \tag{8}$$

Moreover, equation (4) is not very accurate due to to the error between $\boldsymbol{r}_d$ and $\boldsymbol{r}_e$. Thus a vector-valued error function is defined as follows:

$$\epsilon = \boldsymbol{r}_d - \boldsymbol{r}_e. \tag{9}$$

In order to make each element of the error function converges to zero, by utilizing zeroing neural network design formula

$$\dot{\epsilon} = -\mu_1 \epsilon \tag{10}$$

where $\mu_1 \in \mathbb{R}^+$ is a positive parameter to adjust the convergence rate, and $\dot{\boldsymbol{\epsilon}}$ is the derivative of $\boldsymbol{\epsilon}$ with respect to t. By substituting (9) into (10), we can get:

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{r}}_d + \mu_1(\boldsymbol{r}_d - \boldsymbol{r}_e). \tag{11}$$

By using (6) and (11), the fault-tolerant control equation constraints can be easily incorporated into QP1. The equation

(6) thus can be combined together with joint physical constraints and joint velocity constraints into the QP1.

$$\text{minimize } \boldsymbol{\Phi}(t) = \frac{1}{2}\dot{\boldsymbol{\theta}}^T\dot{\boldsymbol{\theta}} \tag{12}$$

$$\text{subject to } \boldsymbol{J}\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{r}}_d + \mu_1(\boldsymbol{r}_d - \boldsymbol{r}_e) \tag{13}$$

$$\boldsymbol{A}\dot{\boldsymbol{\theta}} = 0 \tag{14}$$

$$\boldsymbol{\xi}^- \leqslant \dot{\boldsymbol{\theta}} \leqslant \boldsymbol{\xi}^+. \tag{15}$$

where $\boldsymbol{\xi}^-$ and $\boldsymbol{\xi}^+$ are both n dimensional vectors, representing the lower and upper physical joint limits of the manipulator respectively. Assuming that the joints of the manipulator have speed limits and angle limits below:

$$\begin{aligned} \boldsymbol{\theta}^- &\leqslant \boldsymbol{\theta} \leqslant \boldsymbol{\theta}^+ \\ \dot{\boldsymbol{\theta}}^- &\leqslant \dot{\boldsymbol{\theta}} \leqslant \dot{\boldsymbol{\theta}}^+. \end{aligned} \tag{16}$$

The specific definition of $\boldsymbol{\xi}^-$ and $\boldsymbol{\xi}^+$ are as follows:

$$\begin{aligned} \boldsymbol{\xi}^- &= max(\beta(\boldsymbol{\theta}^- - \boldsymbol{\theta}), \dot{\boldsymbol{\theta}}^-) \\ \boldsymbol{\xi}^+ &= min(\beta(\boldsymbol{\theta}^+ - \boldsymbol{\theta}), \dot{\boldsymbol{\theta}}^+). \end{aligned} \tag{17}$$

Here, β is a parameter used to adjust the size of the feasible region after the joint limit conversion. Now the time-varying fault-tolerance control problem of the redundant manipulator is transformed into a new QP problem. It can be rewritten as the following QP problem (QP2):

$$\text{minimize } \boldsymbol{\Phi}(t) = \frac{1}{2}\dot{\boldsymbol{\theta}}^T\dot{\boldsymbol{\theta}} \tag{18}$$

$$\text{subject to } \boldsymbol{C}\dot{\boldsymbol{\theta}} = \boldsymbol{d} \tag{19}$$

$$\boldsymbol{\xi}^- \leqslant \dot{\boldsymbol{\theta}} \leqslant \boldsymbol{\xi}^+. \tag{20}$$

where matrix $\boldsymbol{C}$ is defined as $\boldsymbol{C} = [\boldsymbol{J}^T, \boldsymbol{A}^T]^T \in \mathbb{R}^{(m+m_f)\times n}$ and vector $\boldsymbol{d}$ is defined as $\boldsymbol{d} = [(\dot{\boldsymbol{r}}_d + \mu_1(\boldsymbol{r}_d - \boldsymbol{r}_e))^T, \boldsymbol{0}^T]^T \in \mathbb{R}^{m+m_f}$.

By using dual decision variables, QP2 can be firstly transformed into a set of linear variational inequalities [32]. A primal-dual equilibrium vector $\boldsymbol{y}^* \in \Omega := \{\boldsymbol{y}|\boldsymbol{y}^- \leqslant \boldsymbol{y} \leqslant \boldsymbol{y}^+\} \subset \mathbb{R}^{n+m+m_f}$ can be obtained, and it has to be obtained that, $\forall \boldsymbol{y} \in \Omega$,

$$(\boldsymbol{y} - \boldsymbol{y}^*)^T(\boldsymbol{Q}\boldsymbol{y}^* + \boldsymbol{z}) \geqslant 0, \tag{21}$$

where the primal-dual vector $\boldsymbol{y}$, its upper bounds and lower bounds are defined as below:

$$\boldsymbol{y} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \boldsymbol{u} \end{bmatrix}, \boldsymbol{y}^- = \begin{bmatrix} \boldsymbol{\xi}^- \\ -\infty \end{bmatrix}, \boldsymbol{y}^+ = \begin{bmatrix} \boldsymbol{\xi}^+ \\ \infty \end{bmatrix}. \tag{22}$$

where $\boldsymbol{u}$ is an auxiliary variable whose initial value is set as $\boldsymbol{0}$ in this paper. The matrix $\boldsymbol{Q}$ and vector $\boldsymbol{z}$ are defined as below:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix}, \boldsymbol{z} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{d} \end{bmatrix}. \tag{23}$$

According to [32], we can transform the QP2 problem into a equation:

$$P(\boldsymbol{y} - (\boldsymbol{Q}\boldsymbol{y} + \boldsymbol{z})) - \boldsymbol{y} = 0. \tag{24}$$

The parameter of the projection function $P(\cdot)$ is a vector. This projection function restricts the lower and upper limits of each element of the input vector $\boldsymbol{y}_{i\times 1}$. Without using this function, the manipulator may be damaged due to that the input values are beyond the joint limits. The specific values of the lower and upper limits of the i-th element are determined by y_i^- and y_i^+ respectively.

$$P(\boldsymbol{y}_{i\times 1}) = \begin{bmatrix} g(y_1) \\ \cdots \\ g(y_i) \end{bmatrix}, g(\boldsymbol{y}_i) = \begin{cases} y_i^- & y_i \leqslant y_i^- \\ y_i & y_i^- \leqslant y_i \leqslant y_i^+ \\ y_i^+ & y_i^+ \leqslant y_i \end{cases} \tag{25}$$

B. Discrete QP Solver

In this subsection, we propose a discrete QP solver to solve the above QP2 problem. According to [21], [33], the error function of (24) can be defined as:

$$\boldsymbol{e}(\boldsymbol{y}) \doteq \boldsymbol{y} - P(\boldsymbol{y} - (\boldsymbol{Q}\boldsymbol{y} + \boldsymbol{z})). \tag{26}$$

Here $\doteq$ denotes the computational assignment operator. Moreover, the following vector can be defined as the search direction to find the zero point of piecewise-linear projection equation (24):

$$\boldsymbol{g}(\boldsymbol{y}) \doteq \boldsymbol{Q}^T\boldsymbol{e}(\boldsymbol{y}) + (\boldsymbol{Q}\boldsymbol{y} + \boldsymbol{z}). \tag{27}$$

With the iteration index $k = 0, 1, 2, \cdots$, if $\boldsymbol{y}^k \notin \Omega$, the following recursive formula can be obtained for the solution of (24):

$$\boldsymbol{y}^{k+1} \doteq P(\boldsymbol{y}^k - \boldsymbol{\rho}(\boldsymbol{y}^k)\boldsymbol{g}(\boldsymbol{y}^k)), \tag{28}$$

where,

$$\begin{aligned} \boldsymbol{\rho}(\boldsymbol{y}) \doteq \frac{\|\boldsymbol{e}(\boldsymbol{y})\|_2^2}{\|(\boldsymbol{Q}^T + \boldsymbol{I})\boldsymbol{e}(\boldsymbol{y})\|_2^2} &\geqslant \frac{\|\boldsymbol{e}(\boldsymbol{y})\|_2^2}{\|(\boldsymbol{Q}^T + \boldsymbol{I})\|_F^2\|\boldsymbol{e}(\boldsymbol{y})\|_2^2} \\ = \frac{1}{\|(\boldsymbol{Q}^T + \boldsymbol{I})\|_F^2} &> 0. \end{aligned} \tag{29}$$

Here, $\|\cdot\|_2$ denotes the two norm of a vector argument. Above is the solver of the linear variational inequalities [33]. The flowchart of the discrete fault tolerance QP solver is shown in Fig. 1. Therefore, solving this equation is equivalent to solving the above QP2 problem. The fault-tolerance control problem of the redundant manipulator can be solved.

For a control task of duration T, its sampling gap is τ. For each gap, we can set the Eq. (26)-(29) to iterate m times. $m \geq 1$, the larger the m, the higher the accuracy of the result. Therefore, the value of m can be appropriately increased under the premise of ensuring real-time calculation. Thus, the computational complexity of our scheme is $O(Tm/\tau)$. In the revised manuscript, the discussion on computational complexity has been added.

C. Theoretical Analysis of QP Solver

In this part, we analyze the feasibility of the discrete QP solver.

1) Convergence Analysis: It can be proved that the QP solver of a redundant manipulator is convergent. That means the solution of $\boldsymbol{y}^{k+1}$ generated by the proposed solver is convergent to the actual solution $\boldsymbol{y}^*$.

Starting with any initial state $\boldsymbol{y}^0$, generated by the QP solver above, the solution sequence $\boldsymbol{y}^k, k = 0, 1, 2, \cdots$, satisfies:

$$\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^*\|_2^2 \leqslant \|\boldsymbol{y}^k - \boldsymbol{y}^*\|_2^2 - \boldsymbol{\rho}(\boldsymbol{y}^k)\|e(\boldsymbol{y}^k)\|_2^2 \ \forall \boldsymbol{y}^* \in \Omega^*, \tag{30}$$

where Ω^* denotes the set of actual solutions. As shown in (29), $\boldsymbol{\rho} > 0$ guarantees the effective solution of piecewise-linear equation (24). That is the sequence $\boldsymbol{y}^k$ converges globally linearly to an optimal solution $\boldsymbol{y}^*$. The proof can be generalized from [33].

2) Stability Analysis: Here we prove the QP solver of a redundant manipulator is stable.

A Lyapunov function candidate is defined as:

$$v_1 = \frac{\|\boldsymbol{y}^k - \boldsymbol{y}^*\|_2^2}{2}. \tag{31}$$

According to [32], the time-derivative of $\boldsymbol{y}$ is:

$$\dot{\boldsymbol{y}}^k = -(\boldsymbol{Q}^T + \boldsymbol{I})\boldsymbol{e}(\boldsymbol{y}^k) \tag{32}$$

Obviously, the Lyapunov function candidate (31) is positive-definite. Then, the time-derivative of v_1 can be obtained as follow:

$$\begin{aligned} \dot{v_1} = \frac{dv}{dt} &= -(\boldsymbol{y}^k - \boldsymbol{y}^*)^T(\boldsymbol{Q}^T + \boldsymbol{I})\boldsymbol{e}(\boldsymbol{y}^k) \\ &\leqslant -\|\boldsymbol{e}(\boldsymbol{y}^k)\|_2^2 \leqslant 0. \end{aligned} \tag{33}$$

Thus v_1 is positive-definite and $\dot{v_1}$ is negative-definite. According to the Lyapunov theory, the model-free scheme is stable.

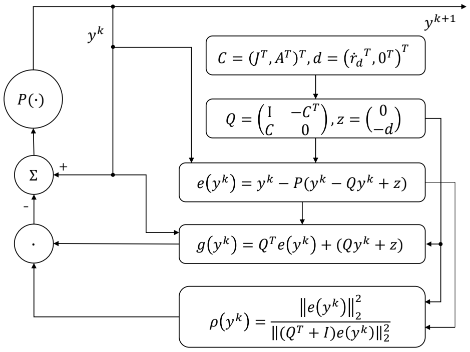

y^k
y^{k+1}
$P(\cdot)$
Σ
+
-
$\cdot$
$C = (J^T, A^T)^T, d = (\dot{r}_d{}^T, 0^T)^T$
$Q = \begin{pmatrix} I & -C^T \\ C & 0 \end{pmatrix}, z = \begin{pmatrix} 0 \\ -d \end{pmatrix}$
$e(y^k) = y^k - P(y^k - Qy^k + z)$
$g(y^k) = Q^T e(y^k) + (Qy^k + z)$
$\rho(y^k) = \frac{\|e(y^k)\|_2^2}{\|(Q^T + I)e(y^k)\|_2^2}$

Fig. 1. Flowchart of the discrete fault tolerance QP solver.

IV. Discrete Model-Free Kinematic Estimator

In this section, we propose a discrete model-free numerical computing solver to achieve the model-free goal.

A. Continuous Kinematic Estimator

In order to solve the QP2 in the condition that without knowing the kinematics model, a vector-valued error function is defined to indicate the error when estimating $\boldsymbol{J}$ used in last section:

$$\boldsymbol{\varepsilon} = \dot{\boldsymbol{r}}_e - \boldsymbol{J}\dot{\boldsymbol{\theta}} \in \mathbb{R}^m. \tag{34}$$

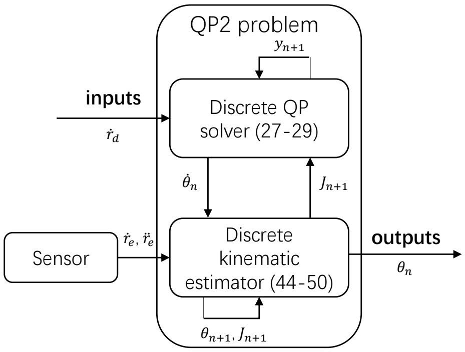


Fig. 2. Schematic diagram of the DMFFTC scheme.

Following the design principle of zeroing neural networks, we can get the formula below:

$$\dot{\boldsymbol{\varepsilon}} = -\mu_2\boldsymbol{\varepsilon} \tag{35}$$

where $\mu_2 \in \mathbb{R}^+$ is a positive parameter to adjust the convergence rate, and $\dot{\boldsymbol{\varepsilon}}$ is the derivative of $\boldsymbol{\varepsilon}$ with respect to t. By substituting (34) into (35), we can get:

$$\ddot{\boldsymbol{r}}_e - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} - \boldsymbol{J}\ddot{\boldsymbol{\theta}} = -\mu_2(\dot{\boldsymbol{r}}_e - \boldsymbol{J}\dot{\boldsymbol{\theta}}). \tag{36}$$

where $\dot{\boldsymbol{J}}$ represents the derivative of $\boldsymbol{J}$ with respect to t. $\ddot{\boldsymbol{r}}_e$ is the derivative of $\dot{\boldsymbol{r}}_e$, which represents the actual acceleration at the redundant manipulator's end-effector. $\ddot{\boldsymbol{\theta}}$ is the derivative of $\dot{\boldsymbol{\theta}}$, which represents the acceleration of each joint of the redundant manipulator. This equation can be rewritten as the following form:

$$\dot{\boldsymbol{J}} = (\ddot{\boldsymbol{r}}_e - \boldsymbol{J}\ddot{\boldsymbol{\theta}} + \mu_2(\dot{\boldsymbol{r}}_e - \boldsymbol{J}\dot{\boldsymbol{\theta}}))\dot{\boldsymbol{\theta}}^{-1} \tag{37}$$

where $\dot{\boldsymbol{\theta}}^{-1}$ here represents the pseudo-inverse of the $\dot{\boldsymbol{\theta}}$ according to the value of m and n. In this equation, $\dot{\boldsymbol{r}}_e$, $\ddot{\boldsymbol{r}}_e$ can be measured by sensors or vision instruments. $\dot{\boldsymbol{\theta}}$ can be obtained by the QP solver in next section.

B. Discrete Kinematic Estimator

In this subsection, we propose a discrete model-free solver based on he continuous model-free scheme derived above.

$\boldsymbol{x}_k$ is defined as the value of $\boldsymbol{x}$ at the time instant $\boldsymbol{t}_k = k\phi$, as well as sampling gap ϕ. By using the vector-form Euler forward formula and vector-form Euler backward formula [34] we have:

$$\begin{aligned} \dot{\boldsymbol{x}}_k &= \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{\phi} + \boldsymbol{O}(\phi), \\ \dot{\boldsymbol{x}}_k &= \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\phi} + \boldsymbol{O}(\phi). \end{aligned} \tag{38}$$

Equations (38) is used to discretize the model-free scheme. Here we define $\kappa = \mu_2\phi$. Thus, the following discrete solver DS-I is obtained:

$$\ddot{\boldsymbol{\theta}}_k \doteq \frac{\dot{\boldsymbol{\theta}}_k - \dot{\boldsymbol{\theta}}_{k-1}}{\phi}, \tag{39}$$

$$\dot{\boldsymbol{J}}_k \doteq (\ddot{\boldsymbol{r}}_{e,k} - \boldsymbol{J}_k\ddot{\boldsymbol{\theta}}_k + \kappa/\phi(\dot{\boldsymbol{r}}_{e,k} - \boldsymbol{J}_k\dot{\boldsymbol{\theta}}_k))\dot{\boldsymbol{\theta}}_k^{-1}, \tag{40}$$

$$\boldsymbol{\theta}_{k+1} \doteq \boldsymbol{\theta}_k + \phi\dot{\boldsymbol{\theta}}_k, \tag{41}$$

$$\boldsymbol{J}_{k+1} \doteq \boldsymbol{J}_k + \phi\dot{\boldsymbol{J}}_k. \tag{42}$$

Similarly, by using Taylor-Zhang discretization formula [35] and the 3-instant backward finite difference formula, we have:

$$\begin{aligned}\dot{\boldsymbol{x}}_k &= \frac{2\boldsymbol{x}_{k+1} - 3\boldsymbol{x}_k + 2\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}}{2\phi} + \boldsymbol{O}(\phi^2),\\ \dot{\boldsymbol{x}}_k &= \frac{3\boldsymbol{x}_k - 4\boldsymbol{x}_{k-1} + \boldsymbol{x}_{k-2}}{2\phi} + \boldsymbol{O}(\phi^2).\end{aligned} \tag{43}$$

Equations (38) is used to discretize the first two steps of the model-free scheme, and (43) to discretize the rest. Thus, the discrete solver DS-II can be derived as follows:

$$\ddot{\boldsymbol{\theta}}_k \doteq \frac{\dot{\boldsymbol{\theta}}_k - \dot{\boldsymbol{\theta}}_{k-1}}{\phi}\ (k \leqslant 2), \tag{44}$$

$$\ddot{\boldsymbol{\theta}}_k \doteq \frac{3\dot{\boldsymbol{\theta}}_k - 4\dot{\boldsymbol{\theta}}_{k-1} + \dot{\boldsymbol{\theta}}_{k-2}}{2\phi}\ (k > 2), \tag{45}$$

$$\dot{\boldsymbol{J}}_k \doteq (\ddot{\boldsymbol{r}}_{e,k} - \boldsymbol{J}_k\ddot{\boldsymbol{\theta}}_k + \kappa/\phi(\dot{\boldsymbol{r}}_{e,k} - \boldsymbol{J}_k\dot{\boldsymbol{\theta}}_k))\dot{\boldsymbol{\theta}}_k^{-1}, \tag{46}$$

$$\boldsymbol{\theta}_{k+1} \doteq \boldsymbol{\theta}_k + \phi\dot{\boldsymbol{\theta}}_k\ (k \leqslant 2), \tag{47}$$

$$\boldsymbol{\theta}_{k+1} \doteq \frac{3}{2}\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1} + \frac{1}{2}\boldsymbol{\theta}_{k-2} + \phi\dot{\boldsymbol{\theta}}_k\ (k > 2), \tag{48}$$

$$\boldsymbol{J}_{k+1} \doteq \boldsymbol{J}_k + \phi\dot{\boldsymbol{J}}_k\ (k \leqslant 2), \tag{49}$$

$$\boldsymbol{J}_{k+1} \doteq \frac{3}{2}\boldsymbol{J}_k - \boldsymbol{J}_{k-1} + \frac{1}{2}\boldsymbol{J}_{k-2} + \phi\dot{\boldsymbol{J}}_k\ (k > 2) \tag{50}$$

where $\dot{\boldsymbol{\theta}}$ used in this solver is obtained from $\boldsymbol{y}$ in QP solver. For the initial values of the above solver, θ_0 is the initial value of the inputs, such as the initial angle of the manipulators' joints. $\dot{\theta}_0$ and $\ddot{\theta}_0$ are set to $\vec{\mathbf{0}} \in \mathbb{R}^n$. As for J_0, its value is determined via motor babbling. That is, independently altering the state of the ith input by a small value $\Delta\theta_i$, the corresponding displacement of manipulator's end-effector Δr_e is measured. Then, the ith column of J_0 can be initialized as $J_0 = [\frac{\Delta r_{e1}}{\Delta\theta_1}, \frac{\Delta r_{e2}}{\Delta\theta_2}, \cdots, \frac{\Delta r_{en}}{\Delta\theta_n}]$. As shown in Fig. 2, by combining the discrete QP solver and kinematic estimator, the DMFFTC scheme can be synthesized for handling the QP2 in Section III.

C. Theoretical Analysis

In this part, we analyze the theoretical analysis of the proposed model-free solver.

1) Convergence Analysis: It can be proved that the discrete model-free solver DS-I and DS-II of a redundant manipulator is convergent.

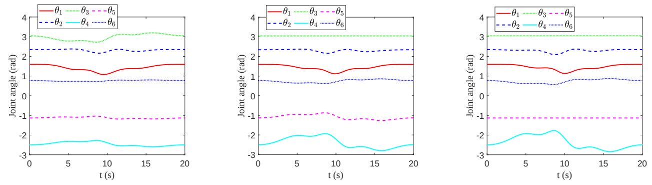
Similar to [36], we can obtain that DS-I model is convergent with $\boldsymbol{O}(\gamma^2)$, i.e., $\boldsymbol{J}_k = \boldsymbol{J}_k^* + \boldsymbol{O}(\gamma^2)$, where $\boldsymbol{J}_k^*$ is the theoretical solution. Thus we have:

$$\|\boldsymbol{J}_k - \boldsymbol{J}_k^*\|_2 = \|\boldsymbol{J}_k^* + \boldsymbol{O}(\gamma^2) - \boldsymbol{J}_k^*\|_2 = \|\boldsymbol{O}(\gamma^2)\|_2. \tag{51}$$

Analogously, simulation error $\|\boldsymbol{J}_k - \boldsymbol{J}_k^*\|_2$ of DS-II scheme is convergent with $\boldsymbol{O}(\gamma^3)$, the proof is complete.

2) Stability Analysis: Here we prove the discrete model-free solver DS-II of a redundant manipulator is stable. A Lyapunov function candidate is defined:

$$\begin{aligned} v_2 &= \frac{\|\ \boldsymbol{\varepsilon}\ \|_2^2}{2} = \frac{\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}}{2}\\ &= \frac{\sum_{i=1}^m \sum_{j=1}^m \boldsymbol{\varepsilon}_{ij}^2}{2} \geqslant 0. \end{aligned} \tag{52}$$



(a) (b) (c)

Fig. 4. The joint angles of the redundant manipulator under different conditions. (a) All joints work normally. (b) The third joint θ_3 motion fails. (c) The third joint θ_3 and the fifth joint θ_5 motion fail.

Obviously, the Lyapunov function (52) is positive-definite. Then, the time-derivative of v_2 is derived as follow:

$$\begin{aligned} \dot{v_2} &= \frac{dv_2}{dt} = \boldsymbol{\varepsilon}^T\frac{d\boldsymbol{\varepsilon}}{dt}\\ &= -\mu_2\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} \leqslant 0. \end{aligned} \tag{53}$$

According to the Lyapunov theory, the DS-I and DS-II scheme are stable.

V. Discrete Joint State Observer

As introduced in (7), matrix $\boldsymbol{A}$ is required to represent the running state of the joints. However, in reality, we can not know in advance when and which joints will fail. Thus, estimating the running state of joints is a necessary task. To do that, we proposed a discrete joint space observer in this section.

At the beginning, we set $m_f = n$ and $\boldsymbol{A} = zeros(n, n)$, assuming that all joints are in good condition. By using a primal-dual neural network [32], we can obtain:

$$\dot{\boldsymbol{y}}_r = \mu_3(\boldsymbol{I} + \boldsymbol{Q}^T)\{P[\boldsymbol{y}_r - (\boldsymbol{Q}\boldsymbol{y}_r + \boldsymbol{z})] - \boldsymbol{y}_r\} \tag{54}$$

where $\boldsymbol{y}_r$ is the dual variable of $\dot{\boldsymbol{\theta}}_r$; $\boldsymbol{\theta}_r$ is the joint vector of redundant manipulator measured by sensors; $\mu_3 \in \mathbb{R}^+$ is a positive parameter to adjust the convergence rate.

Then, we use the differential equation (54) to construct the joint space observer. Firstly, considering some joints of the redundant manipulators may fail, we define a diagonal matrix $\boldsymbol{\Lambda}$ for estimation:

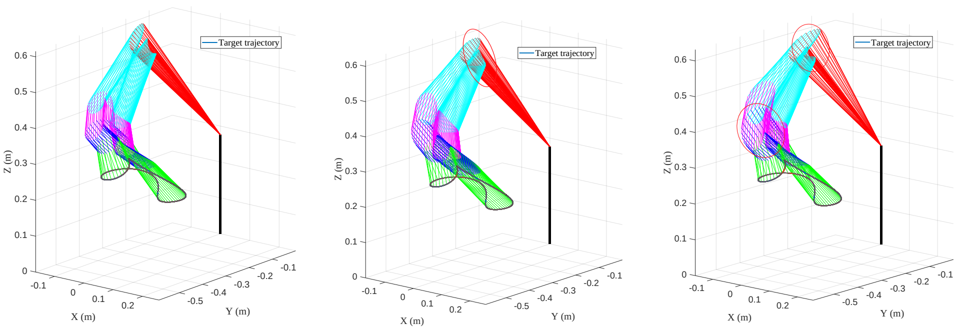
$$\begin{cases} \hat{\boldsymbol{y}}_r = \boldsymbol{\Lambda}\boldsymbol{y}_r\\ \dot{\hat{\boldsymbol{y}}}_r = \boldsymbol{\Lambda}\dot{\boldsymbol{y}}_r \end{cases} \tag{55}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_{n+m} \end{bmatrix} \in \mathbb{R}^{(n+m)*(n+m)}. \tag{56}$$

Similar to matrix $\boldsymbol{A}$, $\boldsymbol{\Lambda}$ can indicate the joint states of the redundant manipulator. The difference is that, when the $i-th$ ($i \leqslant n$) joint fails after a time instant, σ_i will become zero; when the $i-th$ joint works properly, the $\sigma_i \neq 0$ holds. After integrating matrix $\boldsymbol{\Lambda}$, (54) can be rewritten as:

$$\dot{\boldsymbol{y}}_r = \mu_3\boldsymbol{\Lambda}(\boldsymbol{I} + \boldsymbol{Q}^T)\{P[\boldsymbol{y}_r - (\boldsymbol{Q}\boldsymbol{y}_r + \boldsymbol{z})] - \boldsymbol{y}_r\} \tag{57}$$

Through evaluating the value of σ_i, we can detect which or when the joint is failed during the tracking control task. Thus



(a)

(b)

(c)

Fig. 3. The motion states of the redundant manipulator when performing trajectory tracking under different conditions where all the joints and links are visualized at every sampling instant. (a) All joints work normally. (b) The third joint θ_3 motion faults with the joint trajectory illustrated by an ellipse. (c) The third joint θ_3 and the fifth joint θ_5 motion fault with the joint trajectories illustrated by two ellipses.

the dynamic observer can be conducted by using differential equations system below for the redundancy resolution:

$$\begin{cases} \dot{\hat{\boldsymbol{y}}}_r = \hat{\boldsymbol{\Lambda}}\dot{\boldsymbol{y}}_r - k_1(\hat{\boldsymbol{y}}_r - \boldsymbol{y}_r) \\ \dot{\hat{\boldsymbol{\Lambda}}} = k_2\dot{\boldsymbol{y}}_r(\hat{\boldsymbol{y}}_r - \boldsymbol{y}_r)^T \end{cases} \tag{58}$$

where $k_1 > 0$ and $k_2 > 0$ are the convergence scaling parameters, and $\hat{\boldsymbol{y}}_r - \boldsymbol{y}_r$ denotes the error-correction term.

Next, in order to evaluate the convergence properties for the estimated $\hat{\boldsymbol{y}}_r$, we define the following distance variables:

$$\begin{cases} \dot{\tilde{\boldsymbol{y}}}_r = \dot{\boldsymbol{y}}_r - \dot{\hat{\boldsymbol{y}}}_r \\ \dot{\tilde{\boldsymbol{\Lambda}}} = \dot{\boldsymbol{\Lambda}} - \dot{\hat{\boldsymbol{\Lambda}}} \end{cases} . \tag{59}$$

Combining (54)-(58), we have the following distance systems for evaluating convergence of distance variables:

$$\begin{cases} \dot{\tilde{\boldsymbol{y}}}_r = -\tilde{\boldsymbol{\Lambda}}\dot{\boldsymbol{y}}_r - k_1\tilde{\boldsymbol{y}}_r \\ \dot{\tilde{\boldsymbol{\Lambda}}} = k_2\dot{\boldsymbol{y}}_r\tilde{\boldsymbol{y}}_r^T \end{cases} \tag{60}$$

Considering our scheme is discrete, we also discretize the observer system following the same discretization method introduced in Section IV-B:

$$\dot{\tilde{\boldsymbol{y}}}_{r,k} \doteq \tilde{\boldsymbol{\Lambda}}_k\dot{\boldsymbol{y}}_{r,k} - k_1\tilde{\boldsymbol{y}}_{r,k} \tag{61}$$

$$\tilde{\boldsymbol{y}}_{r,k+1} \doteq \tilde{\boldsymbol{y}}_{r,k} + \phi\dot{\tilde{\boldsymbol{y}}}_{r,k},\ (k \leqslant 2) \tag{62}$$

$$\tilde{\boldsymbol{y}}_{r,k+1} \doteq \frac{3}{2}\tilde{\boldsymbol{y}}_{r,k} - \tilde{\boldsymbol{y}}_{r,k-1} + \frac{1}{2}\tilde{\boldsymbol{y}}_{r,k-2} + \phi\dot{\tilde{\boldsymbol{y}}}_{r,k},\ (k > 2) \tag{63}$$

$$\dot{\tilde{\boldsymbol{\Lambda}}}_k = k_2\dot{\boldsymbol{y}}_{r,k}\tilde{\boldsymbol{y}}_{r,k}^T \tag{64}$$

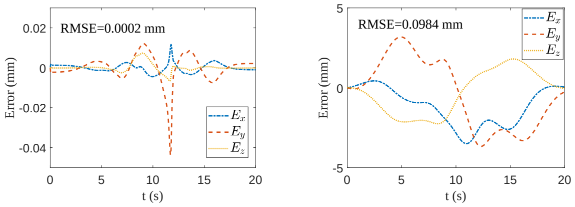
$$\tilde{\boldsymbol{\Lambda}}_{k+1} \doteq \tilde{\boldsymbol{\Lambda}}_k + \phi\dot{\tilde{\boldsymbol{\Lambda}}}_k,\ (k \leqslant 2) \tag{65}$$

$$\tilde{\boldsymbol{\Lambda}}_{k+1} \doteq \frac{3}{2}\tilde{\boldsymbol{\Lambda}}_k - \tilde{\boldsymbol{\Lambda}}_{k-1} + \frac{1}{2}\tilde{\boldsymbol{\Lambda}}_{k-2} + \phi\dot{\tilde{\boldsymbol{\Lambda}}}_k\ (k > 2) \tag{66}$$

The estimated variables $\tilde{\boldsymbol{y}}_r$ and $\tilde{\boldsymbol{\Lambda}}$ should converge to zero as time evolves. Generally speaking, the judgment condition is that, while $\tilde{\boldsymbol{y}}_r$ converges to zero, if $\tilde{\sigma}_i$ of distance matrix $\tilde{\boldsymbol{\Lambda}}$ does not keep at zero, then the $i - th$ joint is detected to be fault. Thus, based on the discrete joint space observer introduced above, the working state of joint can be detected automatically.

VI. Simulations

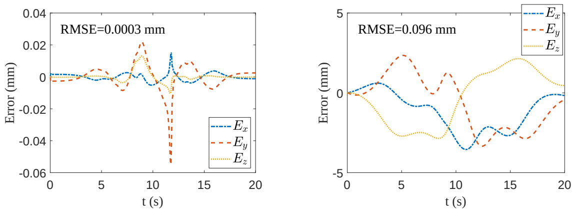
In this section, we implement the DMFFTC scheme derived above through simulations.



(a)

(b)

Fig. 5. Tracking errors of the redundant manipulator with faulty θ_3. (a) Our scheme. (b) Model-based scheme.



(a)

(b)

Fig. 6. Tracking errors of the redundant manipulator with faulty θ_3 and θ_5. (a) Our scheme. (b) Model-based scheme.

A. Settings

In the simulations, the duration of the control task is $T_d = 20$ s. The sampling gap ϕ is set as 0.001 s. The redundant manipulator we use for simulation is "Kinova Jaco2". The initial input is set as $\theta(0) = [1.596, 2.341, 3.045, -2.498, -1.131, 0.77]^T$. The positive parameters μ_1 and μ_2 are set as 100. The desired trajectories $\boldsymbol{r}_d = [dx, dy, dz]^T$ are set as:

$$\begin{aligned} &dx = 3r\cos(2\pi(\sin(0.5\pi t/T_d))^2) + \\ &\quad r\cos(6\pi(\sin(0.5\pi t/T_d))^2) - 4r + initPos(x), \\ &dy = 3r\sin(2\pi(\sin(0.5\pi t/T_d))^2) + \\ &\quad r\sin(6\pi(\sin(0.5\pi t/T_d))^2) + initPos(y), \\ &dz = -2r\sin(2\pi(\sin(0.5\pi t/T_d))^2) - \\ &\quad r\sin(4\pi(\sin(0.5\pi t/T_d))^2) + initPos(z) \end{aligned} \tag{67}$$

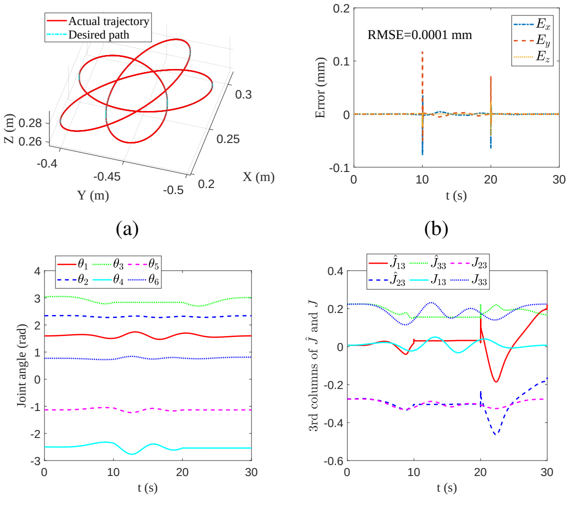
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3149919, IEEE Transactions on Industrial Informatics

where r is used to adjust the size of trajectory and in this case 0.03 m. It is worth noting that although our method can achieve joint fault tolerance, joint faults will narrow the working space of the redundant manipulator due to loss of degrees of freedom.

For given desired trajectories, the redundant manipulator can easily complete the control task in the condition that all the joints work normally, as shown in Fig. 3(a). The root mean squared error (RMSE) is 0.0001 mm, which is a negligible error in a majority of applications. From Fig. 4(a), we can see that all six joints are involved in fulfilling the task, but the changes of joints θ_5 and θ_6 are not significant due to redundancy.

B. One Faulty Joint

The situation that the third joint (θ_3) of the redundant manipulator is considered to be faulty here. As shown in Fig. 3(b), the third joint θ_3 (the joint connecting the red bar and the cyan bar), does not generate any rotation. By using the DMFFTC scheme, the redundant manipulator is able to complete the tracking task with a RMSE of 0.0002 mm as shown in Fig. 5(a), which is quite a negligible error. From Fig. 4(b) we can clearly illustrate that, θ_3 did not rotate at all during the task. To compensate the missing DOF, the rotations of other joints (especially θ_4 and θ_5) have increased slightly.



(a) (b)

(c) (d)

Fig. 7. Situation of normal-faulty transition. (a) The desired path and actual trajectory. (b) Simulation error. (c) The joint angles during the simulation. (d) The third columns of estimated $\hat{\boldsymbol{J}}$ and actual $\boldsymbol{J}$ during the simulation.

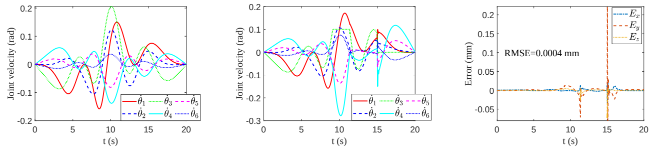
C. Two Faulty Joints

Then, in the case when both the third and the fifth joints of the redundant manipulator fault at the same time, it may not be able to see a obvious difference, but we can still see the control task is well performed from Fig. 3(c). The achieved RMSE is 0.0003 mm as illustrated in Fig. 6(a). As shown in Fig. 4(c), θ_3 and θ_5 does not rotate at all during the task, and the rotations of the remaining normal joints are increased to remedy the missing DOF.

D. Comparison

As far as we know, our scheme is the only model-free scheme for fault tolerant tracking control of redundant manipulators based on recurrent neural networks. Therefore, we compared it with a model-based fault tolerant algorithm in [2] with simulated model errors. Without loss of generality, the DH parameters of the model are added with small biases. i.e., $\alpha_4 = 1.047+\Delta\nu$ and $d_6 = -0.1729+\Delta\nu$ where $\Delta\nu = 0.008$.

As shown in Fig. 5, when θ_3 faults, the root mean squared error (RMSE) of using our method is 0.0002 mm. However, considering the model error, the RMSE of using model-based method is 0.0984 mm, which is much larger than that using our method. Similar results also occur when both θ_3 and θ_5 fault, as shown in Fig. 6. Modeling errors are usually unavoidable for model-based methods, however, thanks to its model-free nature, the DMFFTC scheme can bypass the uncertainties caused by the modeling errors.



(a) (b) (c)

Fig. 8. The velocity simulations curves. (a) The joint velocities in the condition that all joint works well. (b) The joint velocities in the condition that θ_3's velocity is limited. (c) The simulation error in the condition that θ_3's velocity is limited.

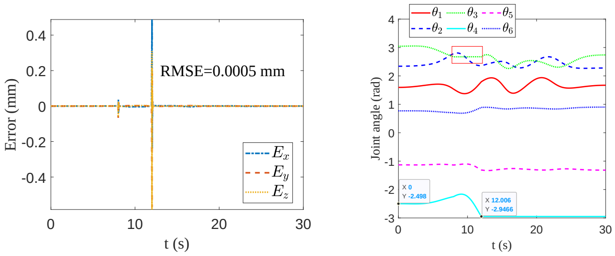
E. Normal-Faulty Transition

To show smoothness of the DMFFTC scheme when coping with the joint-state shifting from normal to fault mode, we carry out an additional simulation to show the tracking performance during this normal-faulty transition. The goal trajectories $\boldsymbol{r}_d = [dx, dy, dz]^T$ are set as:

$$\begin{aligned}
dx &= 2r\sin(6\pi(\sin(0.5\pi t/T_d))^2)- \\
&\quad r\sin(4\pi(\sin(0.5\pi t/T_d))^2) - 4r + initPos(x), \\
dy &= -2r\cos(6\pi(\sin(0.5\pi t/T_d))^2)- \\
&\quad r\cos(4\pi(\sin(0.5\pi t/T_d))^2) + initPos(y), \\
dz &= 2/3r\cos(6\pi(\sin(0.5\pi t/T_d))^2)+ \\
&\quad 1/3r\cos(4\pi(\sin(0.5\pi t/T_d))^2) - r + initPos(z)
\end{aligned} \tag{68}$$

where $r = 0.03$. In this case, the positive parameters μ_1 and μ_2 are set as 200. The duration of the control task is $T_d = 30$ s, when $0 \leq t < 10$ s, no joint is considered to be failed; when $10 \leq t < 20$ s, θ_3 faults; when $20 \leq t \leq 30$ s, θ_4 and θ_5 fault.

As shown in Figs. 7(a) and 7(b), the actual trajectory is basically along with the desired path that RMSE is only 0.0001 mm. Most amount of the errors are concentrated at the transition points. From Fig. 7(c), we can see that θ_3 fails from $t = 10$ s to $t = 20$ s, θ_4 and θ_5 fault after $t = 20$ s. Fig. 7(d) shows the transformation of the third columns of estimated $\hat{\boldsymbol{J}}$ and actual $\boldsymbol{J}$ during the simulation. When $0 \leq t < 10$ s, estimated $\hat{\boldsymbol{J}}$ basically coincides with actual $\boldsymbol{J}$. However,



(a) (b)

Fig. 9. Situation of away from joint reset state transition. (a) Simulation error. (b) The joint angles during the simulation.

controlled by the DMFFTC scheme, the estimated $\hat{\boldsymbol{J}}$ keeps unchanged during $10 \leq t < 20$ s. When $20 \leq t \leq 30$ s, the estimated $\hat{\boldsymbol{J}}$ is very different from the actual $\boldsymbol{J}$ due to other joints' faults.

F. Joint Speed Limit

The simulation above discusses the condition that the joint is completely failed. However, in practice, the joint typically have a degradation over time which ends in a failure at the end. Thus, our scheme also support to constrain the velocity of specified joints to avoid more serious failures. By adjusting the value of $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ in (17), the velocity limits of failed joints can be adjusted.

First, all joints are set to work normally, just as the settings in section VI-A. No velocity limit is set that $\boldsymbol{\xi}^+$ is set as $[\infty, \infty, \infty, \infty, \infty, \infty]^T$ and $\boldsymbol{\xi}^-$ is set as $[-\infty, -\infty, -\infty, -\infty, -\infty, -\infty]^T$ respectively. As shown in Fig. 8(a), θ_3's velocity is the largest one, reaching 0.2 rad at time $t = 10$ s.

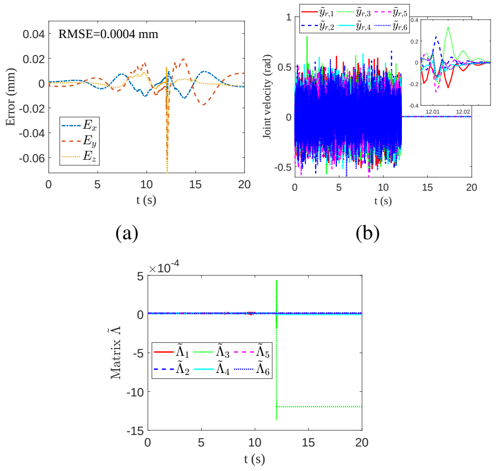
Then, we consider that θ_3 slightly fails at the beginning and completely fails after $t = 15$ s. As shown in Fig. 8(b), θ_3's velocity is maintained at 10 rad around $t = 10$ s, and finally keeps at 0 rad after $t = 15$ s. Meanwhile, in order to make up for the limited θ_3, θ_1 and θ_4's velocities increase significantly and accordingly. The RMSE of the simulation is 0.0002 mm, and the most errors are concentrated at $t = 15$ instant.

G. Transition Far Away From Rest State

In this simulation, the transition happens at a value which is far away from the joint rest state. The simulation settings are the same as Section VI-E. In order to conduct the condition that the joint is far away from the rest state, we firstly set θ_3 fail during the time $8 < t < 12$. During this period, in order to remedy the missing DOF, θ_4 has done a large movement, away from joint reset state. Then we set θ_3 normal, θ_4 fail. As shown in Fig. 9, the transition cause a big error in an instant, but it will not have a big impact on the overall control.

H. Fault Detection by Observer

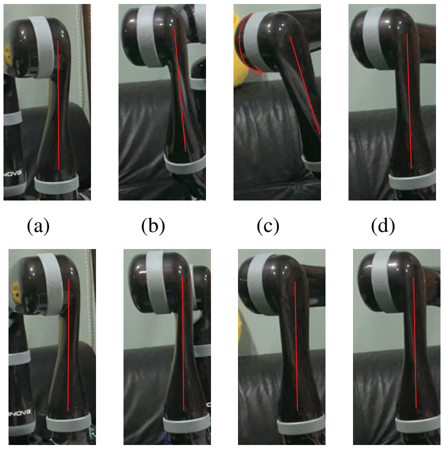
The effectiveness of the discrete joint space observer proposed in Section V is verified in this simulation. The desired trajectories are set as (67) with $r = 0.02$ m. We set all joints of manipulator well at the beginning of simulation, afterwards θ_3 fails at time $t = 12$ s. The parameter μ_3, k_1 and k_2 are set as 100, 200 and 100 respectively. At the beginning of the simulation, all joints of manipulator are set to be working well, then θ_3 is set to fail at time $t = 12$ s. As shown in Fig. 10, at time 0-12 s, $\tilde{\boldsymbol{y}}_r$ constantly converge to zero. When the value of $\tilde{\boldsymbol{\Lambda}}$ keeps at zero (indicating that no joint fails), $\tilde{\boldsymbol{y}}_r$ will be set randomly and the observer continues to keep detecting until the fault happens. After the θ_3 fails at time $t = 12$, σ_3 of $\tilde{\boldsymbol{\Lambda}}$ does not keep at zero, and until $\tilde{\boldsymbol{y}}_r$ converges to zero again, the failure is detected. In this simulation, the large abnormality can be detected within 0.023 seconds. The RMSE of the simulation is 0.0004 mm.



(a) (b)

(c)

Fig. 10. The results of observer when θ_3 fails at time $t = 10$ s. (a) Simulation error. (b) The distance variable $\tilde{\boldsymbol{y}}_r$. (c) The distance variable $\tilde{\boldsymbol{\Lambda}}$.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 12. Local close-up snapshots of θ_3 joint. (a)-(d) θ_3 joint without faulty joints. (e)-(h) θ_3 joint when it faults.

VII. Experiments

In this section, we implement the discrete model-free fault-tolerant control scheme on a physical platform of the redundant manipulator "Kinova Jaco2" to verify its correctness and efficiency.

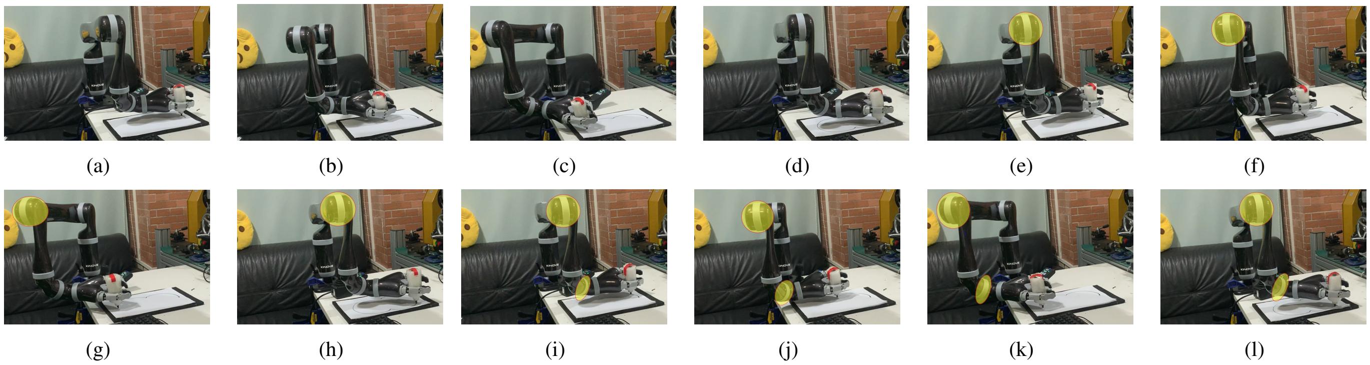


Fig. 11. Snapshots of control results of the redundant manipulator. (a)-(d) all joints work well. (e)-(h) θ_3 fails. (i)-(l) θ_3 and θ_5 fail (with highlighted areas).

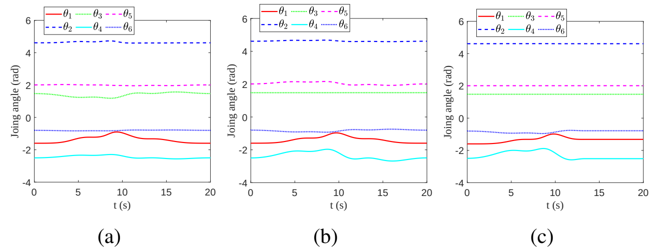


Fig. 13. The joint-angle curves. (a) The joint angles in the condition that all joint works well. (b) The joint angles in the condition that θ_3 fails. (c) The joint angles in the condition that θ_3 and θ_5 fail.

A. Drawing Task

1) Settings: The first experiment is controlling the redundant manipulator to perform the drawing task where the manipulator is holding a pen to draw desired figures. The duration of the control task is set as $T_d = 20$ s and the sampling gap ϕ as 0.001 s. The initial input of the manipulator is set as $\theta(0) = [-1.596, 4.612, 1.474, -2.498, 2.011, -0.801]^T$. The positive parameters $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are set as 100. The desired trajectories $\boldsymbol{r}_d = [dx, dy, dz]^T$ are set as (67). Trajectory scale r is also set to 0.03 m.

The experiment without faulty joints is firstly performed. As shown in Fig. 11(a) to Fig. 11(d), the manipulator complete the task successfully and the corresponding states of the joints are given in Fig. 13(a).

2) One Faulty Joint: To verify the fault tolerance capability, a single joint of θ_3 is set to be not working. The experiment process is shown in Fig. 11(e) to Fig. 11(h), which shows θ_3 being not moving. As we can see, the drawing task task is well performed. To show clearer the results, local close-up snapshots of θ_3 joint are presented in Fig. 12. As shown from Figs. 12(a) to 12(d), the rotation of θ_3 is very obvious under normal condition. On the other hand, governed by the DMFFTC scheme, θ_3 does not generate any rotation as shown from Figs. 12(e) to 12(h). The joint-angle curves are shown in Fig. 13(b) where we can recognize that θ_3 does not rotate at all during the task. In the meantime, rotation amplitudes of θ_4 θ_5 and θ_6 changed evidently.

3) Two Faulty Joints: In this case, the θ_3 and θ_5 are set to fault. Despite the fact that two joints have faulted, the drawing task is completed as shown in Fig. 14(a) to Fig. 14(d), with θ_3 and θ_5 are marked. In Fig. 13(c), we can see that there is no any movement generated in θ_3 and θ_5 during the whole process. In the same time frame, rotation amplitude of θ_4 varied constantly.

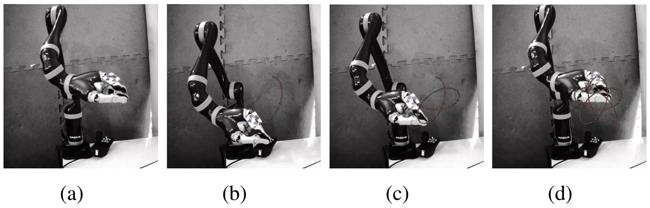


Fig. 14. Snapshots of the trajectory tracking process of the redundant manipulator in the transition simulation.

B. Accuracy Evaluation

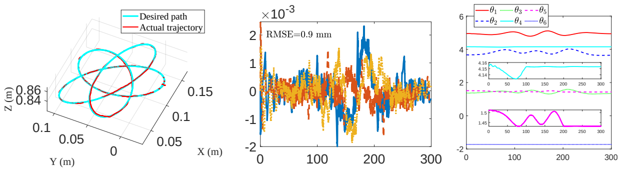
1) Settings: In this section, a visual device is used to evaluate the real-time fault-tolerant tracking performance. The duration and sampling gap ϕ of the control task are set as $T_d = 300$ s and 0.1 s respectively. The initial input of the manipulator is set as $\theta(0) = [4.9467, 3.6651, 1.3740, 4.1542, 1.5102, -1.7237]^T$. Joint θ_4 is set to the fault mode when $t \geq 100$ s, and θ_5 is set to be faulty when $t \geq 200$ s. The desired trajectories $\boldsymbol{r}_d = [dx, dy, dz]^T$ are set as (68). Trajectory scale r is set to 0.025 m.

2) Normal-Faulty Transition: As shown in Figs. 14 and 15(a), under the field of view of the visual device, manipulator successfully complete the tracking task while normal-faulty transition happens. Fig. 15(b) shows that the RMSE of the experiment is 0.9 mm which is an acceptable level for many applications. The state transitions of joint θ_4 and θ_5 are also illustrated in Fig. 15(c).

VIII. Conclusions

In this paper, we propose a discrete model-free fault tolerant tracking control scheme for redundant manipulators. The DMFFTC scheme allows us to solve the redundant manipulator's control problem in the condition that some joints are faulty. This scheme is independent with the kinematic model and can achieve highly accurate tracking control when fault happens in one or more joints. The detailed derivation of the scheme and theoretical analysis are presented. The performances in the cases with one and two faulty joints are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3149919, IEEE Transactions on Industrial Informatics



(a) (b) (c)

Fig. 15. Experiment results of the transition simulation. (a) The desired path and actual trajectory. (b) Simulation error. (c) Joint angles evolution.

verified with simulation and experiments. Comparative studies with the model-based fault tolerant method verified that, in the condition of existing modeling uncertainties, the DMFFTC scheme achieved higher tracking accuracies. In addition, the tracking performance during transition from normal state to faulty state of the joints is verified as well. Lastly, the RMSE of trajectory tracking quantified by the visual device is 0.9 mm. Based on these results, we can definitely conclude that the DMFFTC scheme is competent to realise fault tolerant tracking control without any knowledge about the kinematic model, which brings portability and flexibility for applying the fault tolerant scheme generically.

References

[1] A. Hansson and M. Servin, "Semi-autonomous shared control of large-scale manipulator arms," *Control Engineering Practice*, vol. 18, no. 9, pp. 1069–1076, 2010.

[2] K. Li and Y. Zhang, "Fault-tolerant motion planning and control of redundant manipulator," *Control Engineering Practice*, vol. 20, no. 3, pp. 282–292, 2012.

[3] D. Guo and Y. Zhang, "A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1326–1340, 2012.

[4] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 658–667, 2003.

[5] Y. Chen, J. E. McInroy, and Y. Yi, "Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 680–691, 2003.

[6] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.

[7] D. Bustan, S. K. Sani, and N. Pariz, "Adaptive fault-tolerant spacecraft attitude control design with transient response control," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1404–1411, 2014.

[8] G. Li, D. Song, S. Xu, L. Sun, and J. Liu, "A hybrid model and model-free position control for a reconfigurable manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 785–795, 2019.

[9] Y.-Y. Wu and N. Tan, "Model-less feedback control for soft manipulators with jacobian adaptation," in *2020 International Symposium on Autonomous Systems (ISAS)*, pp. 217–222, 2020.

[10] N. Tan, P. Yu, X. Zhang, and T. Wang, "Model-free motion control of continuum robots based on a zeroing neurodynamic approach," *Neural Networks*, vol. 133, pp. 21–31, 2021.

[11] B. Esmaeili, M. Salim, M. Baradarannia, and A. Farzamnia, "Data-driven observer-based model-free adaptive discrete-time terminal sliding mode control of rigid robot manipulators," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, pp. 432–438, 2019.

[12] H. Wu, S. Jin, C. Yin, J. Zheng, and Z. Hou, "Model free adaptive predictive tracking control for robot manipulators with uncertain parameters," in *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1571–1576, 2021.

[13] B. Liao and W. Liu, "Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators," *Robotica*, vol. 33, no. 10, pp. 2100–2113, 2015.

[14] D. Chen, Y. Zhang, and S. Li, "Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3044–3053, 2018.

[15] D. Chen and Y. Zhang, "Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4385–4397, 2018.

[16] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 6978–6988, 2016.

[17] D. Guo and Y. Zhang, "Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1203–1217, 2014.

[18] R. G. Roberts, H. G. Yu, and A. A. Maciejewski, "Fundamental limitations on designing optimally fault-tolerant redundant manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1224–1237, 2008.

[19] H. Abdi and S. Nahavandi, "Fault tolerance force for redundant manipulators," in *2010 2nd International Conference on Advanced Computer Control*, vol. 2, pp. 612–617, 2010.

[20] Y. Zhang, Z. Yin, H. Huang, L. He, and L. Jin, "Acceleration-level fault-tolerant scheme for redundant manipulator motion planning and control: Theoretics," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6703–6708, 2017.

[21] K. Li, J. Yang, C. Yuan, J. Xu, X. Dai, and J. Luo, "Fault-tolerant motion planning of redundant manipulator with initial position error," in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 533–538, 2018.

[22] Z. Li, C. Li, S. Li, and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7469–7478, 2020.

[23] N. Zhong, X. Li, Z. Yan, and Z. Zhang, "A neural control architecture for joint-drift-free and fault-tolerant redundant robot manipulators," *IEEE Access*, vol. 6, pp. 66178–66187, 2018.

[24] L. Jin, B. Liao, M. Liu, L. Xiao, D. Guo, and X. Yan, "Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network," *Frontiers in Neurorobotics*, 2017.

[25] C. Cheah, C. Liu, and J. Slotine, "Experiments on adaptive control of robots with uncertain kinematics and dynamics," in *Experimental Robotics IX* (M. H. Ang and O. Khatib, eds.), (Berlin, Heidelberg), pp. 57–67, Springer Berlin Heidelberg, 2006.

[26] M. Ahmadipour, A. Khayatian, and M. Dehghani, "Adaptive task-space control of rigid-link robots with uncertain kinematics and dynamics and without acceleration measurements," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–5, 2013.

[27] B. Xiao and S. Yin, "Exponential tracking control of robotic manipulators with uncertain dynamics and kinematics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 689–698, 2019.

[28] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.

[29] C. Yang, G. Peng, L. Cheng, J. Na, and Z. Li, "Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 5, pp. 3282–3292, 2021.

[30] Y. Zhang, S. Chen, S. Li, and Z. Zhang, "Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4909–4920, 2018.

[31] Y. Zhang, S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2126–2132, 2004.

[32] Y. Zhang, "On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems," in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 1351–1356 vol. 2, 2005.

[33] B. He, "A new method for a class of linear variational inequalities," *Mathematical Programming*, vol. 66, pp. 137–144, 1994.

[34] G. Lindfield and J. Penny, "Numerical methods: Using MATLAB," *Numerical Methods: Using MATLAB*, no. D, pp. 1–608, 2018.

[35] Y. Zhang, L. Jin, D. Guo, Y. Yin, and Y. Chou, “Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization,” *Journal of Computational and Applied Mathematics*, vol. 273, no. C, pp. 29–40, 2015.

[36] M. Yang, Y. Zhang, Z. Zhang, and H. Hu, “Adaptive Discrete ZND Models for Tracking Control of Redundant Manipulator,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7360–7368, 2020.

Ning Tan presently is associate professor at Sun Yat-sen University, Guangzhou, China. He received the Ph.D. degree in automation from the Department of Automatic Control and Micro-Mechatronic Systems, Université de Franche-Comté/Franche-Comté Electronics Mechanics Thermal Science and Optics-Sciences and Technologies Institute, Besançon, France. From 2014 to 2018, he held a post-doctoral position and research fellow position at the Singapore University of Technology and Design and National University of Singapore, Singapore. His research interests include soft robotics, bioinspired design, modular and reconfigurable mechanisms, and micro-nano robotics.

Zhaohui Zhong received the B.E. degree from Sun Yat-sen University, China, in 2020. He is currently pursuing the master degree in software engineering at the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His current research interests include neural networks and robotics.

Peng Yu received the B.E. and M.E. degrees from Sun Yat-sen University, China, in 2019 and 2021, respectively. His current research interests include neural networks, robotics, and control theory.

Zhan Li received the B.S. and M.S. degrees from Sun Yat-sen University, Guangzhou, China, in 2009 and 2011, respectively, and the Ph.D. degree from Laboratoire d’Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM)- INRIA, University of Montpellier, Montpellier, France, in 2014. He is currently a Senior Lecturer with the Department of Computer Science, Swansea University, Swansea, U.K. His current research interests include intelligent control of medical/service robotics and biosignal processing. Dr. Li serves as an Editorial Board Member for PLOS One and a Guest Editor for Frontiers in Neurorobotics and Frontiers in Neuroscience.

Fenglei Ni received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in Mechanical Engineering from the Harbin Institute of Technology, Harbin, China, in 1998, 2002, and 2007, respectively. He is currently a Professor with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. His current research interests include the design and control of robotic systems.