2020-12-01

A Fault-Tolerant Method for Motion Planning of Industrial Redundant Manipulator

Li, Z

http://hdl.handle.net/10026.1/15719

A Fault-Tolerant Method for Motion Planning of Industrial Redundant Manipulator

4 authors, including:

Chunxu Li
University of Plymouth
23 PUBLICATIONS   **95** CITATIONS

Some of the authors of this publication are also working on these related projects:

Automated manufacturing processes for glass-reinforced plastic tanks View project

Research into efficient brazing techniques and visual inspection processes for pipe assemblies View project

A Fault-Tolerant Method for Motion Planning of Industrial Redundant Manipulator

Zhan Li, Chunxu Li, Shuai Li, and Xinwei Cao

Abstract—Nowadays industrial redundant manipulators have been playing important roles in manufacturing fields such as welding and assembling by performing repetitive and dull work. Such long-term industrial operations usually require redundant manipulators to keep good working conditions and maintain steadiness of joint actuation. However, some joints of redundant manipulators may fall into fault status after enduring long-period heavy manipulations, causing that the desired industrial tasks can not be accomplished accurately. In this paper, we propose a novel fault-tolerant method with simultaneous fault-diagnose function for motion planning and control of industrial redundant manipulators. The proposed approach is able to adaptively localize which joints run away from the normal state to be fault, and it can guarantee to finish the desired path tracking control even when these fault joints lose their velocity to actuate. Simulation and experiment results on a Kuka LBR iiwa manipulator demonstrate the efficiency of the proposed fault-tolerant method for motion control of the redundant manipulator.

Index Terms—Redundant manipulator, joint failure, motion planning

I. Introduction

Industrial redundant manipulators have been widely applied in manufacturing and service fields by helping various practitioners undertake repetitive and dull jobs which they can not afford. In many critical and special applications (e.g., doing precise assembling of aircraft/automative components, and processing hazardous chemical/nuclear materials), after long-period heavy operations, some joints of these redundant manipulators may fail to keep running as desired due to sudden actuation failure or unexpected joint lock, making path tracking tasks of end-effectors unable to continue or finish accurately. As some joints are falling into failure unpredictably and suddenly, with the fault-tolerant motion planning method utilized, redundant manipulators are expected to regulate end-effectors in the meanwhile to keep ability of accomplishing accurate motion control tasks. Therefore, motion planning with fault tolerance and localization of these fault joints can be a very important issue in redundancy resolution.

In the past decades, in good working conditions, kinematic control of redundant manipulators have received a lot of attentions. By resolving redundancy to seek suitable control

actions in the joint space that lead to a desired motion in its workspace, accurate motion control of redundant manipulators have been implemented under various requirements. Strong coupled nonlinearity issues in kinematic modeling have been handled numerically by pseudoinverse of the robot arms' Jacobian matrix directly [1], as the analytical solution is difficult to find and generalize in the joint level. In order to enhance computational ability in real-time implementation and additionally fulfill secondary performance indices (e.g., joint motion limits, rectifying joint angle drift, and limiting mechanical energy consumption), constrained-optimization based approaches synthesized by parallel-processing methods have been investigated to different performance indices in addition to guaranteed accurate motion control [2]. Some researchers have formulated elegant optimization paradigms. For instance, Kemeny proposed a parameterized form of the kinematic equations null space to solve the constrained optimization [3]. Zanchettin and Rocco developed a null-space optimization algorithm for dual-arm redundancy resolution with dynamically-scaled postures [4]. Xiang *et al.* proposed a general-weighted least-norm method with a new concept of virtual joints for redundant resolution and achieved promising end-effector tracking control performance [5], [6].

For handling with such constrained optimization problems for inverse kinematic control of redundant manipulators, recurrent neural networks have been applied to exhibiting good convergence ability and adaptivity to uncertainties [7], [8], [9], [10]. Many researchers have made progress on this topic. Xia *et al.* proposed a primal dual neural network for online resolving constrained kinematic redundancy in robot motion control [11]. Zhang *et al.* proposed a kinematically resolution subject to joint limits and joint velocity limits for redundant manipulators [2]. Xiao *et al.* proposed an novel acceleration-level motion resolution scheme for repetitive motion control of six-link planar robot manipulator [12]. Chen *et al.* proposed a novel and robust motion scheme to suppress unexpected disturbances for control of mobile robot manipulators [13]. Jin *et al.* reformulated the motion planning scheme for redundant robots to a new quadratic programming problem which can inherently tolerate noises [14]. Guo *et al.* proposed a hybrid redundancy resolution with both weighted velocity and acceleration minimization as co-criteria for robot control [15]. Zhang *et al.* compared three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulator to remedy joint-drift problems [16]. These works have successfully and efficiently resolved inverse kinematics problems with various performance indices satisfied with efficient optimization formulated. Such efficiency still needs to be integrated to

develop fault-tolerant motion planning methods for dealing with various joint failure cases of redundant manipulators in stable industrial operations.

In order to implement accurate kinematic control of industrial redundant manipulators with joint failures, it is also essential to resolve redundancy to seek residual reference joint action that lead to a desired motion in its workspace. In past years, some researchers have proposed fault-tolerant analysis and planning strategies. Pseudoinverse-type methods for fault-tolerant motion resolution approaches have been proposed [17], [18]. Such pseudoinverse-type methods may be not efficient for overcome Jacobian-matrix singularities when joint failures occur. Ben-Gharbia *et al.* analyzed fault tolerance properties of Jacobian matrix for planar manipulators [19], which can enumerate all the possible optimal configurations for resolution. Li *et al.* proposed a fault-tolerant constraint in the optimization criteria for motion planning for redundant manipulators [20], and the coefficient matrix for the constraint is constructed to be sparse with normal and fault states as entries. Zhang *et al.* proposed an acceleration-level fault-tolerant scheme for redundant manipulator motion planning by incorporating a reduced dimension style [21]. Li *et al.* proposed a fault-tolerant motion planning method to make the initial position errors fast converge to zero [22]. Sun and Wang designed nonlinear fault-tolerant motion control for robust compensation of gait training robot system [23].

Towards accurate motion control of redundant manipulators with possible joint failure during the running process, it is still a feasible way to formulate the fault-tolerant motion planning problem through establishing optimization criteria. On the other hand, being aware of which joint may go to failure during motion planning and control in the meantime is also important. To detect which joint transfer to fault state and simultaneously carry on the accurate motion planning and control may more applicable and useful for long-term heavy industrial process. As discussed previously, among the existing methods for dealing with motion planning and control of redundant manipulators [11], [13], [24], [25], [26], [14], [27], most of approaches deal with motion analysis and resolution of redundant manipulators without joint failures considered. However, currently there might be almost no related work on simultaneous fault-tolerant motion planning and fault-diagnose method on redundancy motion control of redundant manipulators.

Motivated by the discussions and considerations above, in order to supplement current kinematics control strategies which mainly tackle with redundant manipulators in normal running states, in this work, we are making breakthroughs to propose a new and novel fault-tolerant motion planning and control method of industrial redundant manipulators with potential joint failure for simultaneously detection of the fault joint(s). Thus it is feasible to guarantee accurate motion control and provide feedbacks on the exact location of fault joint(s) at the same time. The proposed approach is formulated as a constrained optimization problem with a designed observer for joint state monitoring. The primal dual neural network solver is integrated into solution of the constrained optimization problem. The contributions of this paper are summarized as follows:

1) To the best of our knowledge, this paper might be the first work to propose a fault-tolerant motion planning scheme for industrial redundant manipulators with simultaneously detecting/localizing which joint(s) may come into the failure status during the motion control process.

2) The proposed fault-tolerant motion planning scheme is formulated to a convex optimization formulation without additional fault-tolerant equality constraints, which is solved by the primal dual neural network with the fault-tolerant solution set. A dynamic observer for the joint space of the redundant manipulator is constructed to guarantee its convergence and success with proof given. Moreover, a fault-detection algorithm for the joints that possibly fall into failure is established as well.

3) Both simulation results and experiment results on the Kuka industrial redundant robot system demonstrate the efficiency of the proposed fault-tolerant method for achieving desired performance of motion planning which encounters unexpected joint failures, with the accuracy of the motion control satisfied.

II. PROBLEM FORMULATION

In this section, we present the kinematic modeling of industrial redundant manipulators and the formulation of redundancy resolution with joint failures.

A. Kinematics of Industrial Redundant Manipulators

The forward kinematics of the industrial redundant manipulator with n revolute joints can be modelled between the motion of the joint angle and the end-effector, and its relation between Cartesian space and joint space is governed by

$$H(\theta(t)) = \begin{bmatrix} o(t) & r(t) \\ 0 & 1 \end{bmatrix} = H_1(\theta(t))H_2(\theta(t)) \cdots H_n(\theta(t)) \tag{1}$$

where $\theta(t) \in R^n$ denotes the joint space variable, $H(\theta(t)) \in R^{4 \times 4}$ denotes the homogeneous matrix of an end-effector of the redundant manipulator in the universal (base) coordinate, $o(t) \in R^{3 \times 3}$ denotes the orientation matrix of the end-effector, $r(t) \in R^3$ denotes the position vector of the end-effector, and $H_1(\theta(t)), H_2(\theta(t)), \cdots, H_n(\theta(t))$ respectively denotes the homogeneous matrix between adjacent joints at their own local coordinates.

The kinematic chain equation (1) addresses the kinematics characteristics of the redundant manipulator system, and it reflects a nonlinear coupled mapping between joint angular variable θ and position/oritention of the end-effector/end-point. As all the joints of the redundant manipulator are considered to be revolute joints, and estimation and control of the orientation of the end-effector/end-point is not considered in this paper. For its joint angle variable $\theta(t) = [\theta_1(t), \theta_2(t), \cdots, \theta_n(t)] \in R^n$, and its Cartesian space variables $r(t) \in R^3$ in the workspace can be described as the following nonlinear relation from (1):

$$r(t) = f(\theta(t)) \tag{2}$$

where mapping function $f(\cdot)$ reflects geometrical and mechanistic information of the redundant manipulator system. Such

mapping from the manipulator joint space to a its manipulation workspace at the position level and is highly-coupled nonlinear. For such redundancy resolution problem, through solving the nonlinear highly-coupled equations, it is generally to formulate the following inverse kinematics solution

$$\theta(t) = f^{-1}(r(t)) \tag{3}$$

The mapping between joint space variables and Cartesian space variables can be simplified at the velocity level by taking derivations on both sides of $r(t) = f(\theta(t))$ with respect to time as follows

$$\dot{r}(t) = J\dot{\theta}(t)$$

where $J \in R^{3 \times n}$ denotes the Jacobian matrix which is generated by $J = J(\theta(t)) = \partial f / \partial \theta$. These aforementioned derivations are for normal running state of redundant manipulators, i.e., the joint velocity $\dot{\theta}(t)$ do not step into some fault cases such as losing its speed or being locked suddenly from a certain time instant. Therefore, joint failure includes the joint locking or the joint free cases, and it means that the joint velocity can be changed to hold as a zero and/or an arbitrary value. Mathematically, we thus model the failure of joint i as $\dot{\theta}_i(t > t_f) = 0$ for for time instant $t > tf$.

B. Joint Failure of Redundant Manipulator

For industrial redundant manipulators under normal working conditions, the actuation of all the joints are modulated as planned or desired without the failure status encountered like suddenly mutation of joint velocity. However, after enduring long-term repetitive operations, redundant manipulators may undergo breakdown of some joints unexpectedly. Therefore, joint actuation may lose ordered control, causing the resultant joint(s) locked or stuck. In such case, the motion planning formulation can be presented by the following form

$$\text{redundancy resolution} \begin{cases} \theta_d = f^{-1}(r_d) \\ \theta_- \leq \theta \leq \theta_+ \\ \dot{\theta}_- \leq \dot{\theta} \leq \dot{\theta}_+ \end{cases} \tag{4}$$

and

$$\text{joint failure} \begin{cases} \dot{\theta}_{j_1}(t > t_{f1}) = 0 \\ \dot{\theta}_{j_2}(t > t_{f2}) = 0 \\ \quad \vdots \\ \dot{\theta}_{j_m}(t > t_{fm}) = 0 \\ j_1, j_2, \cdots, j_m \in \{1, 2, \cdots, n\} \\ t_{f1}, t_{f2}, \cdots, t_{fm} \in [0 \quad t_e] \end{cases} \tag{5}$$

where θ_d is resolved by inverse kinematics mapping $f^{-1}(\cdot)$ of the redundant manipulator under the joint limit bounds θ_-, θ_+ and joint velocity bounds $\dot{\theta}_-$, $\dot{\theta}_+$, r_d denotes the desired path of the end-effector in Cartesian space, and $t_{f1}, t_{f2}, \cdots, t_{fm}$ denotes the time instants that the $j_1, j_2, \cdots, j_m$th joints begin to fall in failure. From the aforementioned formulations, we can see that when some joints are unexpected to lose to move, the redundancy resolution is still needed to execute to guarantee the desired path r_d be tracked by the end-effector. So joint failure formulation can be seen as an additional constraint for redundancy resolution, and failure recognition and localization construct important parts of fault diagnosis for maintenance and overhaul.

III. PROPOSED FAULT TOLERANT MOTION PLANNING SCHEME

In this section, the proposed fault-tolerant motion planning strategy is addressed with its optimal solution and convergence analysis provided. Furthermore, the joint space observer for the redundant manipulator is established and the joint failure dection/location algorithm is shown correspondingly.

A. Fault Tolerant Motion Planning Scheme

Our aim of this paper is to propose a new fault-tolerant motion planning scheme for the redundant manipulator. As discussed in the previous section, failure joints may be shut down to have no rotation speed output, and this situation can be seen as another constraint for motion planning. However, involvement of another constraint may increase computational complexity. In this paper, we propose the following fault-tolerant redundancy resolution optimization criteria

$$\begin{aligned} \text{minimize} \quad & \|\dot{\theta}\|^2 \\ \text{subject to} \quad & J\dot{\theta} = v_d \\ & \dot{\theta} \in \Omega \end{aligned} \tag{6}$$

where $v_d := \dot{r}_d$ denotes the translational velocity of the end-effector, and $\Omega = \{\dot{\theta} \in R^n, \dot{\theta}_{j_1}(t > t_{f1}) = 0, \cdots, \dot{\theta}_{j_m}(t > t_{fm}) = 0\}$ denotes the fault tolerant set for the joint velocity. Optimization formulation (6) is to minimize the joint velocity to make the kinetic energy as small as possible under the solution set that contains identical equations holding joint velocity zero after some time instants, with an additional constraint that fulfills the end-effector tracking control task. In order to solve such constrained optimization problem, we can develop primal dual neural network solver. According to the design principle of the primal dual neural network [2], we have the following solver for the fault-tolerant redundancy resolution

$$\begin{cases} \epsilon\dot{u} = -u + P_\Omega(u - ku - J^T\lambda) \\ \epsilon\dot{\lambda} = J\dot{\theta} - v_d \end{cases} \tag{7}$$

where $u := \dot{\theta}$ and λ denote the neural network state variables, $\epsilon > 0$ denotes the parameter to scale the convergence, $k > 0$ denotes the scaling parameter, and $P_\Omega(\cdot)$ denotes the linear piecewise projection function with the solution bounds u^- and u^+, i.e.,

$$P_\Omega(u) = \begin{cases} u^+, u > u^+ \\ u, u^- \leq u \leq u^+ \\ u^-, u < u^- \end{cases}$$

B. Convergence of Optimization Solver

To solve the optimization problem (6) for fault-tolerant motion planning efficiently, we equivalently convert it into a set of nonlinear equations. We first review the defined Lagrange function as follows

$$L(u, \lambda) = \|u\|_2^2 / 2 + \lambda^T (Ju - v_d)$$

Due to its convexity of problem, its optimal solution can be equivalently expressed as follows according to the Karush-Kuhn-Tucker (KKT) condition

$$\begin{cases} \frac{\partial L}{\partial \lambda} = 0 \\ P_\Omega\left(u - \frac{\partial L}{\partial u}\right) = u \end{cases}$$

Accordingly, we can have the equations as follows after substituting expressions of $\partial L/\partial u$ and $\partial L/\partial \lambda$ into the equations above

$$\begin{cases} P_\Omega\left(u - ku - J^T\lambda\right) = u \\ Ju - v_d = 0 \end{cases} \tag{8}$$

With the Lagrange function aforementioned, we can obtain the equivalent form of the convex optimization problem as follows

$$\begin{aligned} ku - J^T\lambda &= -\frac{\partial L}{\partial u} \\ Ju - v_d &= 0 \end{aligned} \tag{9}$$

This equivalence is guaranteed by the convexity of the optimization problem (6). Now, we need to see the equivalence of (9) to (8). Note that the above-mentioned expression $\partial L/\partial u \in N_\Omega(u)$ includes the normal cone operator $N_\Omega(u)$. Recall the property on the projection to a normal cone and the second equation in (9), we arrive at the equivalent expression (8). Thus, the solution to optimization problem (6) for solving inverse kinematics resolution with fault tolerance of the redundant manipulator system is identical to the solution to the nonlinear equation set (8).

An analytical solution to the nonlinear equation set (8) can not be obtained. A dynamic recurrent neural network is needed to solve (8), we can have the following theorem on its convergence properties.

Theorem 1: For the robot arm system whose kinematics are generally described by (2), the fault-tolerant primal dual neural network (7) solves its kinematic redundancy resolution with joint failures, and stabilizes the redundantly manipulator to the optimal solution of (6).

Proof. The basic idea of proving Theorem 1 is to convert the neural dynamics into the canonical form presented in [28], [29] and leverage the theoretical results therein to prove the convergence. A similar proof can also be found in [28], [29]. The proof is omitted and interested readers are referred to [28], [29] for the detailed proof. ∎

C. Joint Space Observer and Fault Detection

Although the redundancy resolution solver (7) can deal with the fault-tolerant motion planning issue, which joint(s) is/are falling in failure is still not known. In realistic applications [30], we need to estimate the running state of the joints by knowing the joint is locked or not (i.e., joint velocity is kept being zero in a long period rather than at some instant). By estimating and confirming whether the joint velocity always keep zero or not during a given period, one can readily detect which joints are in failure and which joints are still in a normal/good running state. In this paper, in addition to propose a optimization based fault tolerant recusancy resolution for motion planning, we also propose a failure detection approach

of joints for redundance manipulators as follows. In order to formulate the fault detection of the failure joints, we need to reconstruct the following variant redundancy solver

$$\begin{aligned} \epsilon \dot{u} &= -u + P_\Omega(u - ku - J^T\int(J\dot{\theta} - v_d)dt) \\ &= -u + P_\Omega(u - ku - J^T\int(\dot{r} - \dot{r}_d)dt) \\ &= -u + P_\Omega(u - ku - J^T(r - r_d)) \end{aligned}$$

As $u = \dot{\theta}$ is defined, the equivalent from of the above primal dual neural network solver is

$$\epsilon\ddot{\theta} = -\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d)) \tag{10}$$

As some joints of the redundant manipulator may fail to run, we propose to use the following intermediate-variable equations for further joint velocity estimation

$$\begin{cases} \dot{\hat{\theta}} = \Lambda\dot{\theta} \\ \ddot{\hat{\theta}} = \Lambda\ddot{\theta} \end{cases} \tag{11}$$

where $\dot{\hat{\theta}}$ and $\ddot{\hat{\theta}}$ respectively denote the estimated joint velocity and joint acceleration, and Λ denotes the diagonal matrix, i.e.,

$$\Lambda = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \in R^{n \times n}$$

If some joint (e.g., ith joint) falls into the failure state after a given time instant, then σ_i will become zero. If the ith joint is always in a normal running state, then $\sigma_i \neq 0$ holds. Thus, we have the following simultaneous fault-tolerant and fault-diagnose primal dual neural network solver with the induction matrix Λ equipped

$$\epsilon\ddot{\theta} = \Lambda[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))] \tag{12}$$

According to the fault-tolerant and fault-diagnose solver (12), one can detect which joint is faulty during fault tolerant motion planning by evaluating the value of σ_i. Thus it essentially requires a dynamic observer to estimate such kind of parameters for motion planning, so we construct the auxiliary system as the observer for the redundancy resolution solver as follows

$$\begin{cases} \epsilon\ddot{\hat{\theta}} = \hat{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))] - k_1(\dot{\hat{\theta}} - \dot{\theta}) \\ \epsilon\dot{\hat{\Lambda}} = k_2[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))](\dot{\hat{\theta}} - \dot{\theta})^T \end{cases} \tag{13}$$

where $k_1 > 0$ and $k_2 > 0$ are the convergence scaling parameters, and $\dot{\hat{\theta}} - \dot{\theta}$ denotes the error-correction term. The proposed redundancy resolution solver (12) can make fault-tolerant motion planning and detect the failure joint(s) simultaneously with observer (13) for estimation utilized.

Next, in order to evaluate the convergence properties for the estimated joint variables, we define the following distance variables

$$\begin{cases} \dot{\tilde{\theta}} = \dot{\theta} - \dot{\hat{\theta}} \\ \tilde{\Lambda} = \Lambda - \hat{\Lambda} \end{cases} \tag{14}$$

where $\dot{\hat{\Lambda}}$ denotes the time derivative of estimated matrix $\hat{\Lambda}$ during motion planning.

From equations (10)-(13), we can have the following distance systems for evaluating convergence of the distance variables

$$\epsilon \ddot{\tilde{\theta}} = -\tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))] - k_1\dot{\tilde{\theta}} \qquad (15)$$

and

$$\epsilon \dot{\tilde{\Lambda}} = k_2[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]\dot{\tilde{\theta}}^T \qquad (16)$$

where the estimated variables $\dot{\hat{\theta}}$ and $\hat{\Lambda}$ have to respectively follow the true joint angle velocity $\dot{\theta}$ and failure state matrix Λ finally, and the distances variables $\dot{\tilde{\theta}}$ and $\tilde{\Lambda}$ should converge to zero as time t evolves. For investigating the convergence properties for distance system (15), we provide the following theoretical results with proof.

Theorem 2. For distance system (15), distance variables $\dot{\tilde{\theta}}$ and $\tilde{\Lambda}$ can globally converge to zero as time t evolves, i.e, estimated variable $\dot{\hat{\theta}}$ and converges to the real (resolved) joint angle velocity $\dot{\theta}$.

Proof. For proving the convergence of the distance system

$$\epsilon \ddot{\tilde{\theta}} = -\tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))] - k_1\dot{\tilde{\theta}},$$

we firstly define the Lyapunov function candidate as follows

$$V = \|\dot{\tilde{\theta}}\|^2/2 + \mathrm{tr}(\tilde{\Lambda}^T\tilde{\Lambda})/2k_2 \geq 0$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix, i.e., the sum of all of its diagonal entries.

The time derivative function of Lyapunov V is depicted by

$$\dot{V} = \dot{\tilde{\theta}}^T \ddot{\tilde{\theta}} + \mathrm{tr}(\tilde{\Lambda}^T\dot{\tilde{\Lambda}})/k_2$$

Since we have the following equations according to the distance system equation(15) on $\ddot{\tilde{\theta}}$

$$\ddot{\tilde{\theta}} = -\tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]/\epsilon - k_1\dot{\tilde{\theta}}/\epsilon$$

Therefore, the time derivative of V further becomes

$$\dot{V} = \dot{\tilde{\theta}}^T \frac{-\tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))] - k_1\dot{\tilde{\theta}}}{\epsilon}$$
$$+ \mathrm{tr}(\frac{\tilde{\Lambda}^T[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]\dot{\tilde{\theta}}^T}{\epsilon})$$
$$= -\frac{k_1}{\epsilon}\dot{\tilde{\theta}}^T \dot{\tilde{\theta}} - \frac{1}{\epsilon}\dot{\tilde{\theta}}^T \tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]$$
$$+ \mathrm{tr}(\frac{\tilde{\Lambda}^T[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]\dot{\tilde{\theta}}^T}{\epsilon})$$
$$= -\frac{k_1\|\dot{\tilde{\theta}}\|^2}{\epsilon} \leq 0$$

Evidently, we can see that the Lyapunov function V is positive definite and $\dot{V}$ is negative definite. It indicates that the distance variable $\dot{\tilde{\theta}}$ can globally converge to zero as $t \to +\infty$. As $\dot{\tilde{\theta}} = \dot{\theta} - \dot{\hat{\theta}}$, we can conclude that the estimated $\dot{\hat{\theta}}$ can converge to $\dot{\theta}$ as $t \to +\infty$. The proof is thus complete. ∎

Furthermore, we would like to evaluate the convergence of the second distance system

$$\epsilon \dot{\tilde{\Lambda}} = k_2[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]\dot{\tilde{\theta}}^T.$$

According to the well-known Lasalle invariant set theory, as $\dot{\tilde{\theta}}$ converges to zero which is proved already for Theorem 1, i.e., $\lim_{t \to +\infty} \dot{\tilde{\theta}} = 0$, we can have

$$\lim_{t \to +\infty} -\tilde{\Lambda}[-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))]/\epsilon = 0$$

For the ith fault joint of the redundant manipulator, the primal dual neural network solver can make the ith entry of $-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))$ still converge to zero so as to finish the path tracking control, in order to make the aforementioned equation always hold, the ith entry $\tilde{\sigma}_i$ of $\tilde{\Lambda}$ may not be able to converge to zero. According to the analysis, we would have the following algorithm (Algorithm 1) for detecting the fault joints.

Algorithm 1: Fault joint detection for redundant manipulators

Input : The kinematic model $f(\cdot)$ and Jacobian matrix $J(t)$ of the redundant manipulator, the desired path trajectory r_d of the end-effector, convergence parameter ϵ, k, k_1, and k_2

Output: Distance matrix $\tilde{\Lambda}$

1 Initialize the initial joint angles $\theta(0)$
2 **while** $-\dot{\theta} + P_\Omega(\dot{\theta} - k\dot{\theta} - J^T(r - r_d))$ *converges to zero* **do**
3   **if** $\tilde{\sigma}_i$ *does not converge to zero* **then**
4   The ith joint is detected to be fault
5   **end**
6   **else**
7   There are no faulty joints
8   **end**
9 **end**

IV. SIMULATION RESULTS

In the simulation validation, we firstly examine the performance for the Kuka LBR iiwa industrial manipulator by the proposed method. The DH parameters for joints and links of the Kuka LBR iiwa robot are presented by Tab. I. The Kuka LBR iiwa industrial robot platform system possesses seven links and seven revolute joints. The modeling of the kinematics of the Kuka LBR iiwa industrial robot is given by the transformation matrix between base coordinate and end-effector coordinate

$$T(\theta) = T_1(\theta_1) \cdots T_i(\theta_i) \cdots T_7(\theta_7) \qquad (17)$$

where the Kuka industrial robot's homogeneous transformation matrices are given as follows

$$T_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The desired motion target is respectively planned as a circle path with its radius being 0.20 m and a square path with its side length being 0.20 m in the X-Y planar plane. The convergence parameter ϵ of the primal dual neural network solver for the

Fig. 1. The joint angles θ of the Kuka redundant manipulator with joint failures for circle path tracking control by the proposed fault-tolerant method. Joint 3 falls into the fault state without velocity given anymore after 5 s, joint 5 falls into the fault state without velocity given anymore after 10 s, and joint 7 is locked all the time.
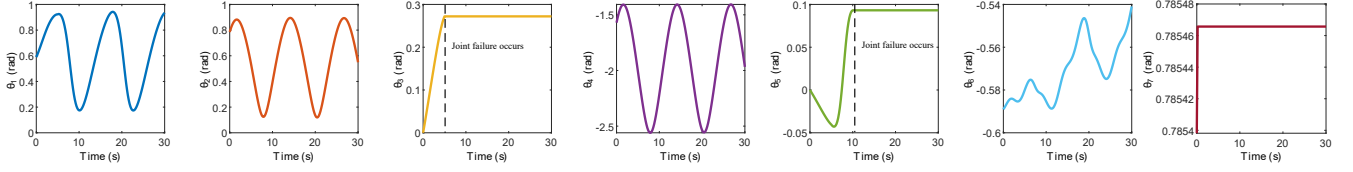

Fig. 2. The joint angles θ of the Kuka redundant manipulator with joint failures for square path tracking control by the proposed fault-tolerant method. Joint 3 falls into the fault state without velocity given anymore after 10 s, joint 6 falls into the fault state without velocity given anymore after 15 s, and joint 7 is locked all the time.
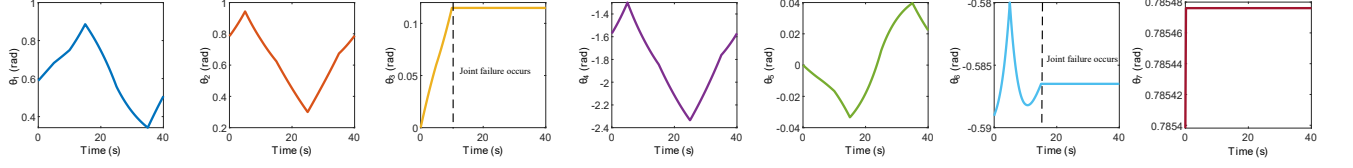
proposed method is chosen as 0.0001. The initial joint angles are set as $\theta(0) = 1.5 \times [\pi/8 \ \pi/6 \ 0 \ -\pi/3 \ 0 \ -\pi/8 \ \pi/6]^T$. The motion control duration is limited to 30 s.

TABLE I
D-H PARAMETERS OF THE KUKA ROBOT PLATFORM.

Link	a_i (m)	α_i (rad)	d_i (m)	θ_i (rad)
1	0	$-\pi/2$	0.340	θ_1
2	0	$\pi/2$	0	θ_2
3	0	$\pi/2$	0.400	θ_3
4	0	$-\pi/2$	0	θ_4
5	0	$-\pi/2$	0.400	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.126	θ_7

A. Path Tracking Performance with Joint Failure

Figs. 1 and 2 show the joint angles θ of the Kuka industrial manipulator for the its end effector to track the desired circle and square paths. For the circle path motion tracking control, joint 3 goes to failure after time instant 5 s with its velocity $\dot{\theta}_3$ being zero afterwards, joint 5 falls to failure after 10 s and then its velocity $\dot{\theta}_5$ becomes zero, joint 7 is always locked from the beginning by keeping its angle constant without any velocity raised during the whole running process, and other joints are in the normal working condition and their angles are resolved by the proposed method. For the square path motion control, joint 3 goes to failure after time instant 10 s with its velocity $\dot{\theta}_3$ being zero afterwards, joint 6 falls to failure after 15 s and then its velocity $\dot{\theta}_6$ becomes zero, joint 7 is always locked from the beginning by letting its angle constant without any velocity raised during the whole running process, and other joints are in the normal working condition and their angles are resolved by the proposed method. Under these circumstances, from the top vertical view, Fig. 3 further shows the circle and square path tracking performances for the Kuka industrial manipulator by the proposed fault-tolerant method with joint failures. In Figs. 3(a) and 3(c), the piecewise straight

(a) Circle path tracking performance

(b) Position error of the end effector for circle path tracking

(c) Square path tracking performance

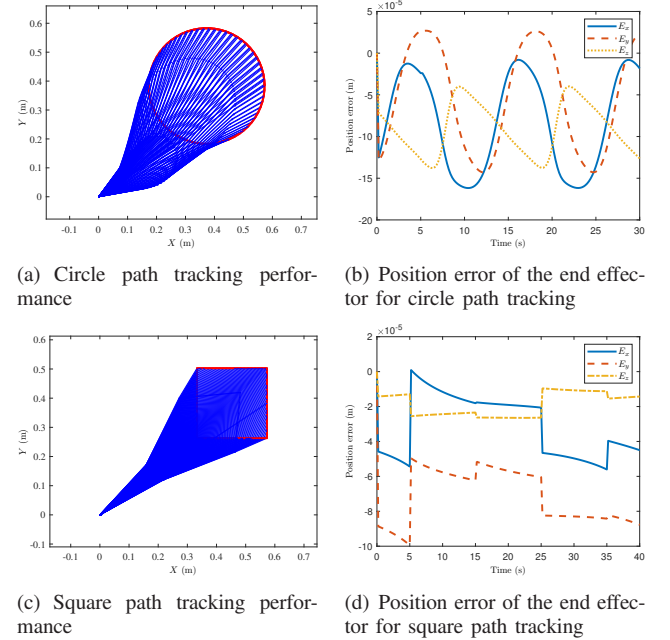(d) Position error of the end effector for square path tracking

Fig. 3. Circle and square paths tracking performance and error evaluation of the Kuka redundant manipulator with joint failures by the proposed fault-tolerant method. The lines in blue represent the 2D projection link body of the 7-link Kuka industrial manipulator, and the curves in red represent the trajectory of the end-effector.

lines in blue represent the 2D projection body of the 7-link Kuka industrial manipulator, and the curves in red represent the trajectory of the end-effector which shows the generated trajectory by the proposed method. We can obviously see that the end-effector of Kuka industrial manipulator can track the desired circle and square paths well with the resolved joint angles by the proposed fault-tolerant method. To evaluate the motion control accuracy, Figs. 3(b) and 3(d) further show the position tracking errors of the end-effector of the Kuka robot, and we can evidently see that the position errors can
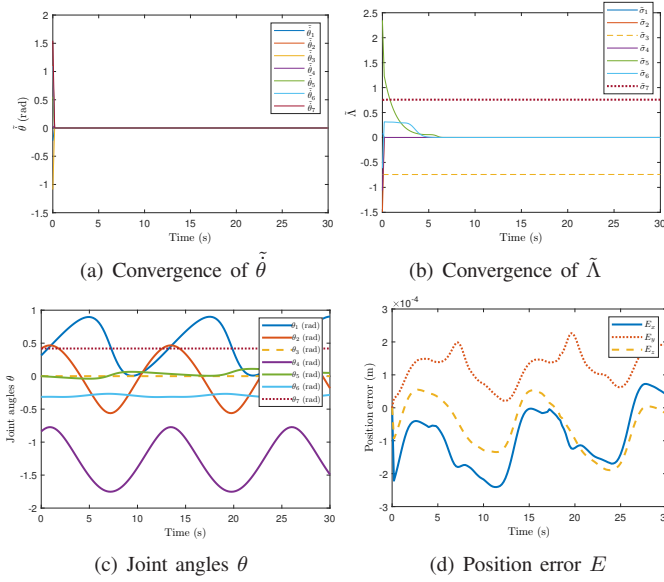
(a) Convergence of $\tilde{\theta}$ (b) Convergence of $\tilde{\Lambda}$

(c) Joint angles θ (d) Position error E

Fig. 4. Convergence of the variables $\tilde{\theta}$ and $\tilde{\Lambda}$ for the Kuka redundant manipulator with failure of joints 3 and 7 for the path tracking task, and resolved joint angles and position tracking error of the end effector are also presented. All of entries of $\tilde{\theta}$ finally converge to zero, indicating that the observer is successfully established. The entries $\tilde{\sigma}_3$ and $\tilde{\sigma}_7$ of $\tilde{\Lambda}$ can not converge to zero, detecting that the failure state occurs in joints 3 and 7.

be lower than 20×10^{-5} m in these two cases. Moreover, as shown in from the path tracking performances and the resolved joint angles of the Kuka industrial manipulator, we can conclude that with joint failures occurring at some time instants, the proposed method can still successfully guarantee the motion control accuracy as expected. All of these simulation results demonstrate the efficiency of the proposed fault-tolerant method for the redundant manipulator for handling such motion planning and control issue with unexpected joint failures.

B. Fault Joint Detection

In addition to validating the proposed fault-tolerant motion planing method for the Kuka industrial manipulator, we further show the simulation results on detection of fault joints afterwards. The Kuka redundant manipulator is guided to track the same circle path with radius being 0.2 m, and joints 3 and joint 7 are locked to be in failure after the initial time instant $t > t_f = 0$ s. Fig. 4 shows the comprehensive performance results of the fault-tolerant proposed method with Algorithm 1 for detecting joint failures. From Fig. 4(a), we see that all the entries of the distance variable $\tilde{\theta}$ converge to zero rapidly with parameters $k = 1$, $k_1 = 10$ and $k_2 = 10$ configured. It validates the converge property addressed in Theorem 2 and indicates that the estimated $\hat{\theta}$ of the designed observer system can successfully track the real joint angles θ. Figs. 4(b) and (c) show the convergence performance for $\tilde{\Lambda}$ and the resolved joint angles θ by the fault-tolerant proposed method, and in the same breath we see that $\tilde{\sigma}_3$ and $\tilde{\sigma}_7$ can not converge to zero and joints 3 and 7 lost their velocity to fall into fault states correspondingly. This phenomena reveals that, if the ith joint may fall into failure unexpectedly, the corresponding ith
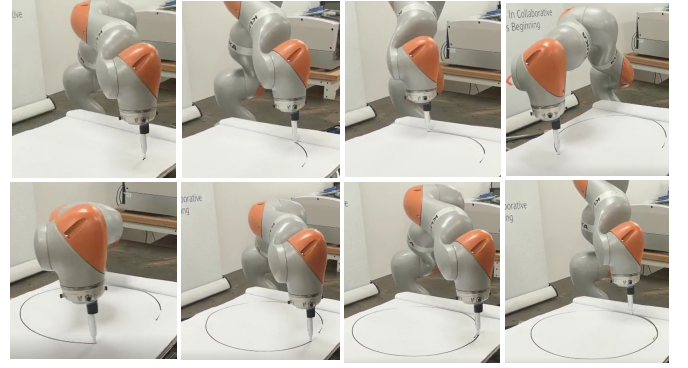

Fig. 5. Experiment snapshots of the Kuka LBR iiwa manipulator for drawing a circle with joint failures planned by the proposed fault-tolerant method.
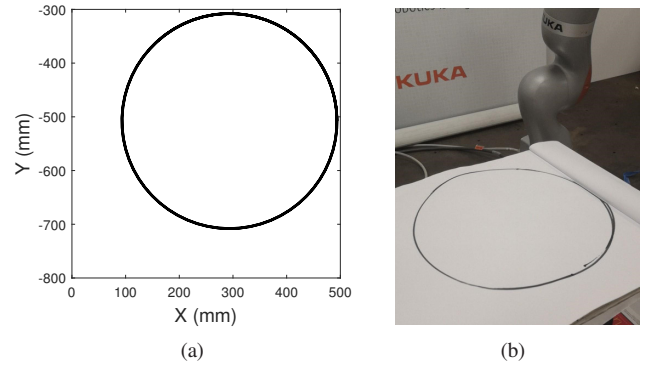

(a) (b)

Fig. 6. Experiment results for the circle tracking task of the Kuka robot with joint failures by the proposed method. Left: recorded path data of the end-effector by the operation system of the Kuka manipulator; right: Drawn path of the end-effector of the Kuka manipulator.

element $\tilde{\sigma}_i$ of $\tilde{\Lambda}$ can not converge to zero. So, one can detect which joint may be in fault through observing which element of $\tilde{\Lambda}$ can not converge to zero. Besides, when achieving the ability of detecting which joint might be fault, the motion control accuracy (less than 3×10^{-4} m for the end-effector) can be guaranteed as well for the proposed method with the designed observer system, which can be clearly seen from Fig. 4(d). All of these results demonstrate the fault detection capability of the proposed method.

V. EXPERIMENT RESULTS

After verifying the proposed method for fault-tolerant motion planning of the Kuka LBR iiwa manipulator from aspect of simulation, in this section, we perform the experimental study based on the Kuka robot platform. The desired path for tracking control is set as a circle with radius being 0.2 m, and the aforementioned optimization solver based on primal dual neural network is utilized to compute the resolved joint angles. The control frequency is set as 10Hz for the Kuka LBR iiwa manipulator and the running time is limited to 30 s for the circle path tracking, and thus 300 time samplings are executed for the control loop. A pen is equipped and fixed at the end-effector to draw the circle during the motion control process.

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.2019.2957186, IEEE Transactions on Industrial Informatics
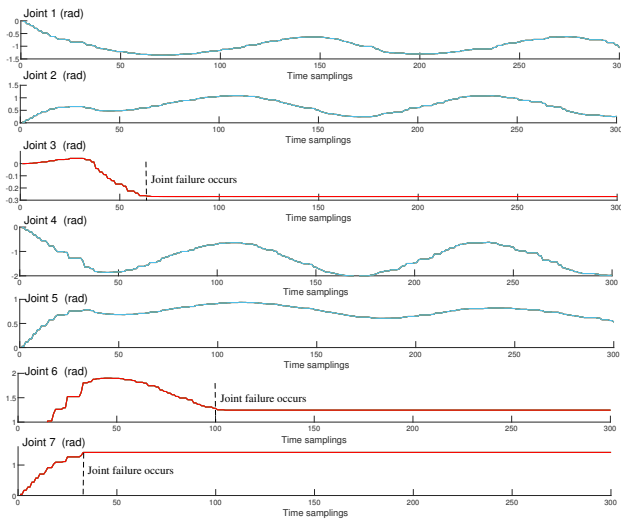
8

Fig. 7. The real sampled joint angles of the Kuka manipulator in the experiment by the proposed method. Joint 3 becomes fault after around 60 time samplings, joint 6 becomes fault after about 100 time samplings, and joint 7 is locked after around 30 time samplings during the entire experiment session.

Fig. 5 shows the motion control process of the Kuka LBR iiwa manipulator for executing the circle path tracking control task, and we can observe that the Kuka robot successfully accomplish the desired circle path drawing. Fig. 6 shows the final path tracking results of the Kuka LBR iiwa manipulator by the proposed fault-tolerant method, and we can see that the path tracking task is well finished. Besides, seen from Fig. 7 that, during the control process of the Kuka LBR iiwa manipulator, joint failures occur in joints 3 and 6 with their velocity becoming zero then, and joint 7 is always locked. Specifically, joint 3 falls into fault after around 60 time samplings, joint 6 falls into fault after around 100 time samplings, and joint 7 is fault after around 30 time samplings during the whole control process. Table II shows the experimental position errors by computing the absolute values of the distance between coordinates (X_d, Y_d) of 10 desired path points and coordinates (X, Y) of 10 actual path points, and the corresponding average position error is $(6.27\pm2.57, 6.00\pm3.56)\times10^{-3}$ m. Together with the path tracking performance as shown in Fig. 6, we can conclude that, even with these joints being fault in unexpected time instants, the Kuka manipulator can still keep to run to finish the path tracking motion control task based on the proposed method. All of these experiment results demonstrate the efficiency of the proposed method for fault-tolerant motion planning of the redundant manipulator.

VI. CONCLUSION

Industrial redundant manipulators have been playing important roles in manufacturing and service industry such as welding and assembling by performing long-time repetitive and dull work nowadays. Such long-period heavy industrial operations generally require redundant manipulators to always keep good working conditions and maintain steadiness of their joint actuation. However, there are always some joints of redundant manipulators fall into the fault state after enduring such long-period heavy jobs and it may degrade motion control performance. A new simultaneous fault-tolerant and fault-detection method for motion planning of redundant manipulators is proposed. The proposed method is able to localize the fault joints which do not move unexpectedly. Simulation and experiment results on a Kuka LBR iiwa manipulator demonstrate the efficiency of the proposed fault-tolerant method for motion control of the redundant manipulator with promising motion control accuracy. Future works can involve tracking control of the orientation of the end-effector to the desired poses and consider developing new fault-tolerant and fault-diagnose approaches for handling with other types of joint failure such as unruled joint velocity cases.

TABLE II
EXPERIMENTAL POSITION ERRORS WITH THE COORDINATES OF 10
SAMPLED DESIRED PATH POINTS AND 10 SAMPLED ACTUAL PATH POINTS.

Point #	(X_d, Y_d) (mm)	(X, Y) (mm)	Position error (m)
1	(148, -645.5)	(152.9, -635.5)	$(4.9, 10)\times10^{-3}$
2	(198.2, -683.9)	(204.8, -675.9)	$(6.6, 8)\times10^{-3}$
3	(290.9, -708.1)	(300, -700)	$(9.1, 8.1)\times10^{-3}$
4	(497.3, -467.1)	(489.8, -468.2)	$(7.5, 1.1)\times10^{-3}$
5	(418, -351.1)	(422.8, -342.2)	$(4.8, 8.9)\times10^{-3}$
6	(446.7, -379)	(457.8, -377.2)	$(11.1, 1.8)\times10^{-3}$
7	(349.1, -306.1)	(347.1, -315)	$(2, 8.9)\times10^{-3}$
8	(283.5, -300.7)	(287.7, -307.8)	$(4.2, 7.1)\times10^{-3}$
9	(110.8, -435.1)	(104.7, -440.7)	$(6.1, 5.6)\times10^{-3}$
10	(100, -500)	(93.35, -499.5)	$(6.45, 0.5)\times10^{-3}$

REFERENCES

[1] 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, July 2018.

[2] 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, May 2003.

[3] Z. Kemeny, "Redundancy resolution in robots using parameterization through null space," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 777–783, Aug 2003.

[4] A. M. Zanchettin and P. Rocco, "Dual-arm redundancy resolution based on null-space dynamically-scaled posture optimization," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 311–316.

[5] J. Xiang, C. Zhong, and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 660–669, Aug 2010.

[6] P. Chen, J. Xiang, and W. Wei, "A unified weighted least norm method for redundant manipulator control," *International Journal of Advanced Robotic Systems*, vol. 13, no. 19, pp. 1–10, 2016.

[7] W. S. Tang and J. Wang, "A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 1, pp. 98–105, Feb 2001.

[8] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y. Li, "Neural-dynamic-method-based dual-arm cmg scheme with time-varying constraints applied to humanoid robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3251–3262, Dec 2015.

[9] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2243–2254, Oct 2017.

[10] Y. Li, S. Li, and B. Hannaford, "A model-based recurrent neural network with randomness for efficient control with applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2054–2063, April 2019.

[11] Y. S. Xia, Gang Feng, and Jun Wang, "A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 54–64, Feb 2005.

[12] L. Xiao and Y. Zhang, "Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 906–914, May 2013.

[13] 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, Sep. 2018.

[14] 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, Nov 2016.

[15] 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, Oct 2014.

[16] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1423–1434, June 2017.

[17] 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, Aug 2003.

[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, Oct 2008.

[19] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "A kinematic analysis and evaluation of planar robots designed from optimally fault-tolerant jacobians," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 516–524, April 2014.

[20] 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.

[21] 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*, Oct 2017, pp. 6703–6708.

[22] 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)*, May 2018, pp. 533–538.

[23] P. Sun and S. Wang, "Robust fault-tolerant compensation tracking control for omni-directional rehabilitative training walker," in *2013 IEEE International Conference on Mechatronics and Automation*, Aug 2013, pp. 575–580.

[24] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, Nov 2014.

[25] Z. Hou, L. Cheng, and M. Tan, "Multicriteria optimization for coordination of redundant robots using a dual neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1075–1087, Aug 2010.

[26] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3812–3821, Sep. 2018.

[27] Y. Zhang, L. He, S. Li, D. Chen, and Y. Ding, "Zeroing dynamics based motion control scheme for parallel manipulators," *Electronics Letters*, vol. 53, no. 2, pp. 74–75, 2017.

[28] X.-B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 426–431, March 2003.

[29] X.-B. Gao and L.-Z. Liao, "A neural network for monotone variational inequalities with linear constraints," *Physics Letters A*, vol. 307, no. 2, pp. 118 – 128, 2003.

[30] M. A. Costa, B. Wullt, M. Norrlof, and S. Gunnarsson, "Failure detection in robotic arms using statistical modeling, machine learning and hybrid gradient boosting," *Measurement*, vol. 146, pp. 425 – 436, 2019.