CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

A sparsity-based method for fault-tolerant manipulation of a redundant robot

Zhan Li[1], Chunxu Li[2,*] , Shuai Li[3,*], Shuo Zhu[2] and Hooman Samani[2]

[1]Computer Science Department, Swansea University, Swansea, UK, [2]School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth, UK, and [3]College of Engineering, Swansea University, Swansea, UK
*Corresponding authors. E-mails: chunxu.li@plymouth.ac.uk; shuai.li@swansea.ac.uk
Zhan Li and Chunxu Li are the co-first authors

Abstract

As an important part of the manufacturing industry, redundant robots can undertake heavy and tough tasks, which human operators are difficult to sustain. Such onerous and repetitive industrial manipulations, that is, positioning and carrying, impose heavy burdens on the load bearing for redundancy robots' joints. Under the circumstances of long-term and intense industrial operations, joints of redundant robots are conceivably to fall into functional failure, which may possibly cause abrupt joint lock or freeze at unknown time instants. Therefore, task accuracy by end-effectors tends to diminish considerably and gradually because of broken-down joints. In this paper, a sparsity-based method for fault-tolerant motion planning of redundant robots is provided for the first time. The developed fault-tolerant redundancy resolution approach is defined as L1-norm based optimization with immediate variables involved to avoid discontinuity in the dynamic solution process. Meanwhile, those potential faulty joint(s) can be located by the designed fault observer with the proposed fault-diagnosis algorithm. The proposed fault-tolerant motion planning method with fault diagnosis is dynamically optimized by resultant primal dual neural networks with provable convergence. Moreover, the sparsity of joint actuation by the proposed method can be enhanced by around 43.87% and 36.51%, respectively, for tracking circle and square paths. Simulation and experimental findings on a redundant robot (KUKA iiwa) prove the efficacy of the developed defect tolerant approach based on sparsity.

1. Introduction

Nowadays, redundant robots have been widely applied to ease the burdens of workers or reduce labor costs in many areas of the manufacture industry [1]. With the help of redundant robots, which persist in undertaking repetitive and heavy industrial operations over long periods, practitioners' productive forces can be emancipated on large scales. After excessive industrial manipulations, some joints of redundant robots might unexpectedly lapse into fault states such as joint lock or joint freeze, causing joint control tasks unable to continue accurately. In this situation, the manipulation performance of end-effectors can be significantly degraded. However, as these joints are falling into unpredictable failure suddenly, the end-effectors of redundant robots are still expected to possess the ability to track desired paths with tolerant accuracy. Fault-tolerant motion planning solutions can be used to self-repair/recover from joint failures [2] or be integrated in advance of failures [3] in the motion planning category. The purpose of the former is to change the robot's mobility strategy in the event of a failure, allowing the robot to compensate for the malfunction and complete its work without the need for hardware repair. After robot manipulators fail to adjust for the motion of the locked joints, the motion of the healthy joints must be replanned [4]. The object of the latter situation is to retain the robot in a configuration that ensures good performance in the event of a failure. Even after an arbitrary joint failure, the robot should be able to reach all task sites for point-to-point activities. This may be achieved by ensuring that none of the robot's joints extends beyond the ranges of the self-motion manifolds assigned to each task point.

In order to guarantee accurate motion tracking of redundant robots that encounters a joint failure, generating the residual joint angular variables from the desired path of the end-effector in its workspace is still essential and seems the only manner for redundancy resolution [5]. In this case, joint angular variables of fault joints may not be directly involved in the redundancy resolution without any additional constraints. It is different from the way of processing in good working status for redundant robots, that is, motion planning and control of end-effectors in the workspace can be done through directly seeking control actions of all the joints in the way of redundancy resolution [6, 7, 8], and motion planning and control of redundant robots in normal working conditions have already achieved great success [9] with many efficient approaches have been developed under different scenarios [10, 11]. Some academics have developed fault-tolerant analysis and motion planning methodologies in the past several years in order to find realistic solutions for redundant robots to overcome the difficulties of generalizing analytical solutions from Cartesian space to joint space. Pseudoinverse-based fault-tolerant redundancy resolution algorithms have been developed [12, 13], but the conventional pseudoinverse-based methods might be not efficient to deal with Jacobian matrices' local singularities from fault joints. As a representative analysis work, Ben-Gharbia et al. analyzed Jacobian matrices of planar manipulators under different joint fault cases [14], by enumerating all conceivable optimum resolution configurations from the perspective of mechanisms. Constrained optimization-based methodologies for fault-tolerant synthesis have been developed to improve computational capabilities for real-time implementation while also meeting other performance criteria. For instance, Xie et al. [15] analyzed different joint failure probabilities to optimally generate the fault-tolerant Jacobian matrix of redundant robots for kinematics modeling. Li et al. [16, 17] reformulated a unified optimization problem for motion planning of redundant robots, with additional constraints which contain joint lock/freeze information. In order to improve solution efficiency with the elimination of unnecessary constraints in optimization, Li et al. [18] proposed a fault-tolerant motion planning method for redundant robots by redefining a new solution set including joint failure information.

Fault-diagnosis methods and their applications in numerous industrial processes and systems have generated successful findings during the previous four decades. Fault diagnosis can be broadly divided into model-based fault diagnosis and signal-based fault diagnosis, according to ref. [19]. Models of the robots must be accessible in order to use model-based methodologies, which may be obtained using either physical principles or systems identification techniques. Instead of using specific input−output models, signal-based approaches use observed signals [19]. In realistic application cases, redundant robots may more frequently encounter joint failure situations. However, current motion planning and redundancy resolution works are mostly focused on redundant robots in normal working conditions. Furthermore, among the existing fault-tolerant motion planning methods based on optimization paradigms for handling redundant robots with joint failures, all of the approaches constructed optimization formulation in a nonsparse manner [16, 18], which might induce the residual joint variables still at the full level of actuation and result in kinetic energy out of gauge. In addition, being aware of which joint goes into failure simultaneously is also important for promptly maintaining the working conditions of redundant robots [19, 20].

Sparsity is a term used in AI inference and machine learning to describe a matrix of numbers that contains numerous zeros or values that have no substantial influence on a calculation [21]. For years, researchers have attempted to extract as many unnecessary parameters from the model as possible – all while maintaining AI's remarkable accuracy. In ref. [22], the authors proposed an improved RRTConnect algorithm based on sparse dead point saved strategy to improve the speed of motion planning, and their experimental studies show the strong robustness. Polvara et al. [23] employed the sparse reward for the proposed Deep Q-Networks to be the high-level navigation policy of the landing problem of an unmanned aerial vehicle, the simulation works validated their proposal with great robustness. Bascetta et al. [24] developed a closed-form dynamic model of flexible manipulators, where its performance was to improve by leveraging the sparsity of matrices for real-time control purposes. All of these works have shown their advantages in high speed computing and robustness by integrating the

sparsity into robotics fields; however, to the best of our knowledge, there is presently almost no relevant work on a sparsity-based fault-tolerant strategy for motion planning of redundant robots with defect diagnostics. In order to supplement current motion planning and redundancy resolution strategies on redundant robots, which are primarily considered in normal running states, we are attempting to make breakthroughs in this paper by proposing a sparsity-based fault-tolerant motion planning (SFTMP) with the ability to diagnose redundant robots with possible joint failure. The developed method is expressed as a constrained $L1$ optimization problem with a custom diagnostics detector for joint fault states. This paper's contributions are summarized as follows:

1) This is the first paper to present the SFTMP strategy for redundant robots that includes concurrently recognizing and localizing the defective joint(s).

2) A dynamic fault-diagnosis paradigm is proposed to detect which joint(s) can be falling into a fault state with theoretical results on convergence presented.

3) Simulation and experiment findings on a KUKA iiwa robot platform with unanticipated locked/frozen joints indicate the efficiency of the proposed joint fault tolerance control scheme optimized by sparsity, with task accuracy of motion planning satisfied and joint motion sparsity increased.

2. Problem formulation

The motion of an end-effector of a redundant robot is reflected by a forward kinematics chain model with joint angular variable θ as the input. In the normal working conditions, for a redundant robot whose joints are not falling into failure situations during the entire motion process, its coordinate data $r(t) \in R^m$ in Cartesian space can be expressed as a coupled-nonlinear relationship in the workspace shown below [18]

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

where $\theta(t) \in R^n$ denotes its joint angle variables, and $f(\cdot)$ denotes the nonlinear mapping function array $f(\cdot)$ which reflects the redundant robot's geometrical and mechanistic profiles. Correspondingly, the optimized joint angles are produced by solving the following inverse kinematics problem [18]:

$$\theta(t) = f^{-1}(r(t)) \tag{2}$$

where $f^{-1}(\cdot)$ denotes the inverse kinematics mapping function array. In such normal working conditions, each joint angle $\theta_i(t)$ with $i \in \{1, 2, \cdots, n\}$ does not encounter failures so there are no failure-related solution sets or constraints appeared in the joint resolution process. The connection between values in joint space and Cartesian space, respectively, ought to be turned into velocity values by deriving across both halves of (1) regarding the time t in order to resolve the inverse kinematic problem in a dynamic manner that is associated with variations in joint velocity:

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

the Jacobian matrix is denoted as $J \in R^{m \times n}$. The aforementioned derivations are aiming at normal manipulation of a redundant robot, for instance, the joint motion velocity $\dot{\theta}(t) \in R^n$ does not affect any fault scenarios like being locked/frozen or increased/decreased abruptly from certain time instants.

When some joint(s) fall(s) into the failure status, the joint velocity variables $\dot{\theta}(t) \in R^n$ may become uncontrolled by the servo controllers and actuators or are even difficult to predict through encoders during motion planning and control. In this situation, such abnormal working conditions have impacts on getting the right $\dot{\theta}(t) \in R^n$ to generate the exact position vector to track the desired path $r_d(t) \in R^m$ of the end-effector, and the desired tracking accuracy might be not be achieved based on the velocity

kinematics equation only. This is due to that the fault joint velocities exert a new restriction for inverse solving velocity kinematics equation, that is, [18],

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

where some joint velocity variables $\dot{\theta}'(t) \subset \dot{\theta}(t) \in R^n$ encounter that $\dot{\theta}_j(t), \cdots, \dot{\theta}_{j+q}(t)$ with $j \in \{1, 2, \cdots, n\}, q \in \{1-j, 2-j, \cdots, n-j+1\}$ become arbitrary values after time instant $t \geq t_f$. Additionally, as the input for the Jacobian matrix $J = J(\theta)$ is θ which is integrated by $\dot{\theta}$, the Jacobian matrix is also evidently perturbed in joint failure situations, which makes the joint resolution becoming more complicated.

In this paper, we consider the *j*th joint velocity $\dot{\theta}_j(t_{fj}) = 0$ represents that the *j*th joint goes to failure after time instant t_{fj}. The type of joint failure is joint lock/freeze. The joint resolution formulation with such joint lock/freeze situation can be described generally as follows:

$$
\text{Joint failure}
\begin{cases}
\text{Velocity kinematics } J\dot{\theta}(t) = \dot{r}_d(t) \\
\dot{\theta}_j\left(t_{fj}\right) = 0 \\
\quad\vdots \\
\dot{\theta}_{j+q}\left(t_{fq}\right) = 0 \\
t_{fj}, \cdots, t_{fq} \in [0 \quad T]
\end{cases}
\tag{3}
$$

where some joints like joints $j, \cdots, j+q$ may lose their velocity and become locked or frozen at certain time instants $t_{fj}, \cdots, t_{fq}$ throughout $[0 \quad T]$. From the motion planning and control formulation with joint lock/freeze, the identical equations on the joint velocity $\dot{\theta}_j\left(t_{fj}\right) = 0, \cdots, \dot{\theta}_{j+q}\left(t_{fq}\right) = 0$ may appear suddenly with different time instants and disturb solution of velocity kinematics equations $J\dot{\theta}(t) = \dot{r}_d(t)$. Another issue is that we have to know which identical equation(s) appear(s) by constructing fault-diagnosis functions for monitoring.

3. SFTMP with fault diagnosis

3.1. Fault-tolerant motion planning in a sparsity perspective

In order to fulfill the same types of motion control tasks as the cases in normal working conditions for redundant robots with fault joints, in this work, we propose to make the joint motion resolution by taking advantage of the sparsity of joint actuation. Inspired by the sparse representation in optimization [25, 26] - the output of L_1 norm produces a sparse model that can be used for feature selection while also preventing overfitting, the following L_1 norm ($\|\cdot\|_1$) based optimization is proposed as follows [18]:

$$
\begin{aligned}
\text{minimize} \quad & \|\dot{\theta}\|_1 \\
\text{subject to} \quad & J\dot{\theta} = \dot{r}_d \\
& -\eta \leq \dot{\theta} \leq \eta \\
& \dot{\theta} \in \Omega_{FT}
\end{aligned}
\tag{4}
$$

where $\eta \in R^n$ denotes the joint velocity boundary variable, and $\Omega_{FT} = \{\dot{\theta} \in R^n | \dot{\theta}_{j1}(t > t_{fj1}) = 0, \dot{\theta}_{j2}(t > t_{fj2}) = 0, \cdots, \dot{\theta}_{jm}(t > t_{fjm}) = 0\}$ denotes the solution set for the fault-tolerant redundancy resolution. The index $-\|\dot{\theta}\|_p, 0 < p < 2$ can be used to evaluate the sparsity of the joints $-\|\dot{\theta}\|_p, 0 < p < 2$, where $\|\cdot\|_p$ denotes the *p* norm of a vector.

As directly computing the partical derivative of the Langrange function that is associated the L_1 objective function may result in discontinuity on the dynamic equations, the optimization paradigm based on (4) may not be efficient. In order to remedy this, a new immediate variable w is involved to equivalently reformulate the original optimization as (5), which keeps the convexity and adopts an additional constraint between the joint angular variable and the immediate variable. To this end, we

present a novel equivalent optimization formulation to make the optimization issue (4) well handled by the primal dual neural network

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} w_i \\
\text{subject to} \quad & J\dot{\theta} = \dot{r}_d \\
& -\eta_i \le \dot{\theta}_i \le \eta_i \\
& -w_i \le \dot{\theta}_i \le w_i \\
& \dot{\theta} \in \Omega_{FT}
\end{aligned}
\tag{5}
$$

where w_i is a vector which belongs to $[w_1, w_2, \cdots, w_i, \cdots, w_n]$, denotes the newly constructed variable to be optimized, and dynamically adjusts the new boundaries of $\dot{\theta}_i$.

The new optimization problem above can be further equivalently transformed into the following optimization formation:

$$
\begin{aligned}
\text{minimize} \quad & l^T w \\
\text{subject to} \quad & J\dot{\theta} = \dot{r}_d \\
& -\eta \le \dot{\theta} \le \eta \\
& -w \le \dot{\theta} \le w \\
& \dot{\theta} \in \Omega_{FT}
\end{aligned}
\tag{6}
$$

where

$$
l = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \text{ and } w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}
$$

Optimization problem (6) can be further rewritten as follows:

$$
\begin{aligned}
\text{minimize} \quad & l^T w \\
\text{subject to} \quad & J\dot{\theta} = \dot{r}_d \\
& \dot{\theta} \in \overline{\Omega}
\end{aligned}
\tag{7}
$$

where a new solution set has been defined $\overline{\Omega} = \{\dot{\theta} \in \Omega_{FT} \bigcap \Omega\}$ with $\Omega = \{-\eta \le \dot{\theta} \le \eta$ and $-w \le \dot{\theta} \le w\}$. An optimizing formula (7) is prepared to solve the sparse optimizing problem of L_1-norm based on (4) by inserting a w variable to simultaneously limit the joint velocity $\dot{\theta}$.

For the purposes of optimization problem (7) solving, the following Lagrange function has been defined:

$$
L = l^T w + \lambda^T \left(J\dot{\theta} - \dot{r}_d \right)
$$

The following partial derivative equations can then be obtained by differentiating the above Lagrange function with respect to the unknown variables $z = \begin{bmatrix} \dot{\theta}^T & w^T \end{bmatrix}^T$

$$
\begin{cases}
\dfrac{\partial L}{\partial z} = \begin{bmatrix} J^T \lambda \\ l \end{bmatrix} \\[4ex]
\dfrac{\partial L}{\partial \lambda} = J\dot{\theta} - \dot{r}_d
\end{cases}
\tag{8}
$$

Then, based on the basic design method of the primal dual neural network for solving convex optimization issues, we have the following unique primal dual neural network solver for fault-tolerant sparse optimization of redundancy resolution:

$$
\begin{cases}
\epsilon \dot{z} = -z + P_{\overline{\Omega}} \left\{ z - Kz - \begin{bmatrix} J^T \lambda \\ l \end{bmatrix} \right\} \\
\epsilon \dot{\lambda} = J\dot{\theta} - \dot{r}_d
\end{cases}
\tag{9}
$$

where the solution set cone $\overline{\Omega} = \bigcup_{i=1}^{n} \overline{\Omega}_i$ with $\overline{\Omega}_i = z_i$ with boundaries $-\eta_i \leq \dot{\theta}_i \leq \eta_i$ and $-w_i \leq \dot{\theta}_i \leq w_i$, $\epsilon > 0$ denotes the network convergence scaling parameter, and $K \in R^{2n \times 2n}$ denotes a coefficient diagonal matrix with all of its entries k being nonnegative. For the newly presented $P_{\overline{\Omega}}(\cdot)$ linear projection features with a newly divided solution set $\overline{\Omega}$, we obtain

$$
P_{\overline{\Omega}}(z) = \bigcup_{i=1}^{n} P_{\overline{\Omega}_i}(z_i)
\tag{10}
$$

and we can then extend its subparts by the following way [18]:

$$
P_{\overline{\Omega}_i}(z) =
\begin{cases}
\begin{bmatrix} \dot{\theta}_i \\ w_i \end{bmatrix}, |\dot{\theta}_i| \leq \eta_i \text{ and } |\dot{\theta}_i| \leq w_i \\[2ex]
\begin{bmatrix} \eta_i \\ w_i \end{bmatrix}, \dot{\theta}_i \geq \eta_i \text{ and } w_i \geq \eta_i \\[2ex]
\begin{bmatrix} -\eta_i \\ w_i \end{bmatrix}, \dot{\theta}_i \leq -\eta_i \text{ and } w_i \geq \eta_i \\[2ex]
\begin{bmatrix} \eta_i \\ \eta_i \end{bmatrix}, w_i \leq \eta_i \text{ and } \dot{\theta}_i \geq -w_i + 2\eta_i \\[2ex]
\begin{bmatrix} -\eta_i \\ \eta_i \end{bmatrix}, w_i \leq \eta_i \text{ and } \dot{\theta}_i \leq w_i - 2\eta_i \\[2ex]
\begin{bmatrix} (\dot{\theta}_i + w_i)/2 \\ (\dot{\theta}_i + w_i)/2 \end{bmatrix}, |w_i| \leq \dot{\theta}_i \leq -w_i + 2\eta_i \\[2ex]
\begin{bmatrix} (\dot{\theta}_i - w_i)/2 \\ (-\dot{\theta}_i + w_i)/2 \end{bmatrix}, w_i - 2\eta_i \leq \dot{\theta}_i \leq -|w_i| \\[2ex]
\begin{bmatrix} 0 \\ 0 \end{bmatrix}, |\dot{\theta}_i| \leq -w_i
\end{cases}
\tag{11}
$$

As compared with the previous work [18], a novel linear piece-wise projector feature with a newly proposed solution set $\overline{\Omega}$ is developed in this paper. The new $\overline{\Omega}$ solution expands the previous three divided solution subsets with a further five. During these operations, the primary dual neural network solution can keep its convergence capacity.

Theorem 1. *For motion planning of the redundant robot with joint lock/freeze which described by (3), the proposed sparse optimization solver (9) replaces the kinematics calculus and even optimizes the solution of (4).*

Proof. Because the developed L_1 sparse optimization problem is convex, the optimal solution may be achieved by solving the following equations, regarding the well-known Karush−Kuhn−Tucker method [27]

$$\begin{cases} \dfrac{\partial L}{\partial \lambda} = 0 \\ P_{\overline{\Omega}}\left(z - \dfrac{\partial L}{\partial z} \right) = z \end{cases}$$

i.e.,

$$\begin{cases} P_{\overline{\Omega}}\left(z - Kz - \begin{bmatrix} J^T\lambda \\ l \end{bmatrix} \right) = z \\ J\dot{\theta} - \dot{r}_d = 0 \end{cases} \tag{12}$$

The cone operator $N_{\overline{\Omega}}(z)$ contains $\partial L/\partial z \in N_{\overline{\Omega}}(z)$. The equivalent expression (12) may be obtained using the property on the linear projection to the cone [28, 29]. The solution to the nonlinear equation (12) is the same as the answer to the optimization issue (4) for the redundant robot's fault-tolerant manipulation scheduling. It is difficult to derive an analytical solution directly for the nonlinear problem (12). As a result, a primal dual neural network must be built to tackle the problem (12), with the Lyapunov stability theory proving its convergence. The complete evidence is excluded here, and interested readers can consult theoretical literature [28, 29]. □

3.2. Fault joint diagnosis

Now we have addressed that the sparse optimization (9) is able to plan the manipulation of the redundant robot with fault tolerance functions under the joint lock/freeze situation. Although such motion planning can be achieved with a new sparse solution paradigm, another issue still needs to be solved: which joints are at fault. Through observing and recording the joint velocity, the dynamic states of joints can be used to confirm whether the joints are locked or not. In this subsection, in addition to proposing the SFTMP method for redundant robots, a fault-diagnosis paradigm is also provided simultaneously as an enhanced comprehensive solution. In order to formulate the dynamic fault-diagnosis model for potential fault joints, firstly the following variant sparse optimization solver is constructed as:

$$\begin{aligned} \epsilon\dot{z} &= -z + P_{\overline{\Omega}}\left(z - Kz - \Xi \int v\,dt \right) \\ &= -z + P_{\overline{\Omega}}\left(z - Kz - \begin{bmatrix} J^T \\ I \end{bmatrix} \int \begin{bmatrix} J\dot{\theta} - v_d \\ \dot{\alpha} \end{bmatrix} dt \right) \\ &= -z + P_{\overline{\Omega}}\left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \end{aligned}$$

where

$$\Xi = \begin{bmatrix} J^T \\ I \end{bmatrix} \text{ and } v = \begin{bmatrix} J\dot{\theta} - v_d \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \dot{r} - \dot{r}_d \\ \dot{w} \end{bmatrix}$$

The variant sparse optimization solver above can be further rewritten as

$$\epsilon\dot{z} = -z + P_{\overline{\Omega}}\left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \tag{13}$$

Next, for monitoring the joints whether are locked or not at certain time instants, we construct the following dynamic observer equations for estimation of joint velocity:

$$
\begin{cases}
\hat{z} - \Lambda z = 0 \\
\dot{\hat{z}} - \Lambda \dot{z} = 0
\end{cases}
\tag{14}
$$

where $\hat{z}$ denotes the estimated joint velocity, and Λ which is a diagonal matrix with its element σ_j being either zero or a non-zero scalar denotes the diagnosis matrix. For example, when the jth joint falls into the locked state from some time instant, $\sigma_j = 0$ will appear. When the jth joint is in a normal state, $\sigma_j \neq 0$ can keep existing. Therefore, we establish the following sparse optimization solver with fault-diagnosis ability for fault-tolerant motion planning:

$$
\epsilon \dot{z} = \Lambda \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right]
\tag{15}
$$

Through the dynamic estimator (15), we can readily find which joint is being locked by knowing $\sigma_i = 0$ or $\sigma_i \neq 0$. At this stage, we construct the auxiliary system equations for the sparse optimization solver as follows:

$$
\begin{cases}
\epsilon \dot{\hat{z}} = \hat{\Lambda} \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right] - K_1 \left(\hat{z} - z \right) \\[2mm]
\epsilon \dot{\hat{\Lambda}} = K_2 \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right] \left(\hat{z} - z \right)^T
\end{cases}
$$

where K_1 and K_2 are the new convergence scaling parameter matrices which are also diagonal with their elements being $k_1 > 0$ and $k_2 > 0$, respectively.

In order to get the diagnosis results, the convergence of the estimated joint velocity variables should be investigated. Under these considerations, the following distance variables are defined as $\tilde{z} = z - \hat{z}$ and $\tilde{\Lambda} = \Lambda - \hat{\Lambda}$ to evaluate whether the estimated variables can converge to the real variables.

Next, combining all aforementioned equations, the form of distance system formulas can be written as follows:

$$
\epsilon \dot{\tilde{z}} = -\tilde{\Lambda} \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right] - K_1 \tilde{z}
\tag{16}
$$

and

$$
\epsilon \dot{\tilde{\Lambda}} = K_2 \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right] \tilde{z}^T
\tag{17}
$$

where the estimated values $\hat{\Lambda}$ and $\hat{z}$ are expected to track the corresponding real values during the fault-diagnosis process, respectively, which means that the distance variables $\tilde{\theta}$ and $\tilde{\Lambda}$ can converge to zero in the steady state [18]. To substantiate this point, the sequential theories are presented.

Theorem 2. *Distance parameters $\tilde{z}$ and $\tilde{\Lambda}$ in the distance system formulas (16) can globally converge to zero, which means that estimated value $\hat{z}$ converges to the real value z, which is synthesized by sparse optimization solver (9).*

Proof. Let us see the first distance system equation in fault-tolerant motion planning

$$
\epsilon \dot{\tilde{z}} = -\tilde{\Lambda} \left[-z + P_{\overline{\Omega}} \left(z - Kz - \begin{bmatrix} J^T(r - r_d) \\ w \end{bmatrix} \right) \right] - K_1 \tilde{z},
$$

first of all, the Lyapunov candidate function is defined as

$$
V = \|\tilde{z}\|^2 / 2 + k_2^{-1} \mathrm{tr} \left(\tilde{\Lambda}^T \tilde{\Lambda} \right) / 2 \geq 0
$$

where $\mathrm{tr}(\cdot)$ indicates the square matrix trace that calculates the sum of all the diagonal elements.

The temporal Lyapunov function derivative V is determined

$$\dot{V} = \tilde{z}^T \dot{\tilde{z}} + \text{tr}\left(\tilde{\Lambda}^T \dot{\tilde{\Lambda}}\right) k_2^{-1}$$

As (16) is obtained previously, we have

$$\dot{\tilde{z}} = -\tilde{\Lambda}\left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right]/\epsilon - K_1 \tilde{z}/\epsilon$$

where we define

$$\beta = \begin{bmatrix} J^T (r - r_d) \\ w \end{bmatrix}$$

The temporal Lyapunov function derivative V thus further expands

$$\dot{V} = \tilde{z}^T \frac{-\tilde{\Lambda}\left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right] - K_1 \tilde{z}}{\epsilon}$$
$$+ \text{tr}\left(\frac{\tilde{\Lambda}^T \left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right] \tilde{z}^T}{\epsilon}\right)$$
$$= -\frac{k_1}{\epsilon} \tilde{z}^T \tilde{z} - \frac{1}{\epsilon} \tilde{z}^T \tilde{\Lambda}\left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right]$$
$$+ \text{tr}\left(\frac{\tilde{\Lambda}^T \left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right] \tilde{z}^T}{\epsilon}\right)$$
$$= -\frac{k_1 \|\tilde{z}\|^2}{\epsilon} \leq 0$$

Thus, we would now see that the V function of Lyapunov is positive and its temporal derivative $\dot{V}$ is negative. According to the well-known Lyapunov stability theory, the distance parameters $\tilde{z}$ is able to globally converge to zero while $t \to +\infty$. Since $\tilde{z} = z - \hat{z}$, the estimated $\hat{z}$ can converge to θ as $t \to +\infty$ in the first distance system (16). The proof is thus complete. $\square$

For the second distance system (17) in joint failure diagnosis, we would further have the following theoretical results.

Theorem 3. *The joint lock/freeze status of the redundant robot system can be detected during motion planning, provided that there exist(s) non-zero element(s) $\tilde{\sigma}_j$ in matrix of $\tilde{\Lambda}$ in the second distance system (17).*

Proof. For the second distance system

$$\epsilon \dot{\tilde{\Lambda}} = K_2 \left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right] \tilde{z}^T,$$

according to Theorem 2, $\tilde{z}$ converges to zero, that is, $\lim_{t \to +\infty} \tilde{z} = 0$, applying Lasalle invariant set theory, we would have

$$\lim_{t \to +\infty} -\tilde{\Lambda}\left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right]/\epsilon = 0$$

For the sake of fault-tolerant manipulation of the redundant robot, the sparse optimization solver guarantees the jth coefficient of $-z + P_{\overline{\Omega}}(z - Kz - \beta)$ converges to zero to achieve finishing path tracking. To satisfy

$$-\tilde{\Lambda}\left[-z + P_{\overline{\Omega}}(z - Kz - \beta)\right]/\epsilon = 0$$

the jth coefficient $\tilde{\sigma}_j$ of $\tilde{\Lambda}$ should not converge to zero. That is to say, once $\tilde{\sigma}_j \neq 0$ exists, then the corresponding joint is found to be locked/frozen. Otherwise, no joint(s) can be found to be falling into the fault state. $\square$

Algorithm 1. Fault-diagnosis algorithm of recognizing locked/frozen joint(s) of the redundant robot.

Input   : Forward kinematic equation $f(\cdot)$; Jacobian matrix J; desired path r_d; parameters ϵ,
            K, K_1 and K_2; joint velocity boundary η; initial joint angle $\theta(0)$
Output: Matrix $\tilde{\Lambda}$

1 **while** $-z + P_{\overline{\Omega}}(z - Kz - \beta)$ *converges to zero* **do**
2   **if** $\tilde{\sigma}_j \neq 0$ **then**
3   Prompt: the jth joint is locked/frozen!
4   **end**
5   **else**
6   Prompt: No joints are found to be locked/frozen.
7   **end**
8 **end**
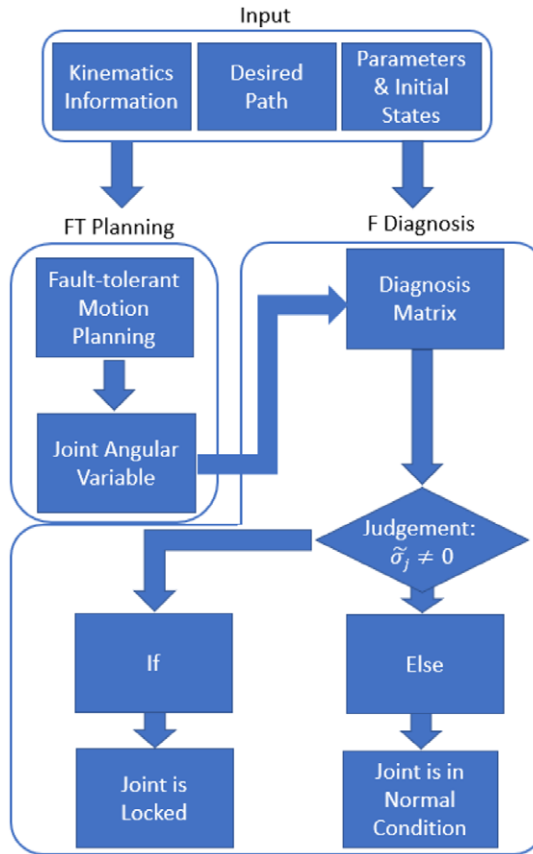
Figure 1. *Flowchart of SFTMP.*

According to Theorems 2 and 3 with related analysis, the following fault-diagnosis algorithm (Algorithm 1) is developed for recognizing the locked/frozen joint(s). A flowchart can be found in Fig. 1 to illustrate the entire presented algorithm.

Table I. *D−H parameters for the KUKA iiwa robot.*

Link	a_i(m)	α_i(degree)	d_i(m)
1	0	−90°	0.340
2	0	90°	0
3	0	90°	0.400
4	0	−90°	0
5	0	−90°	0.400
6	0	90°	0
7	0	0°	0.126
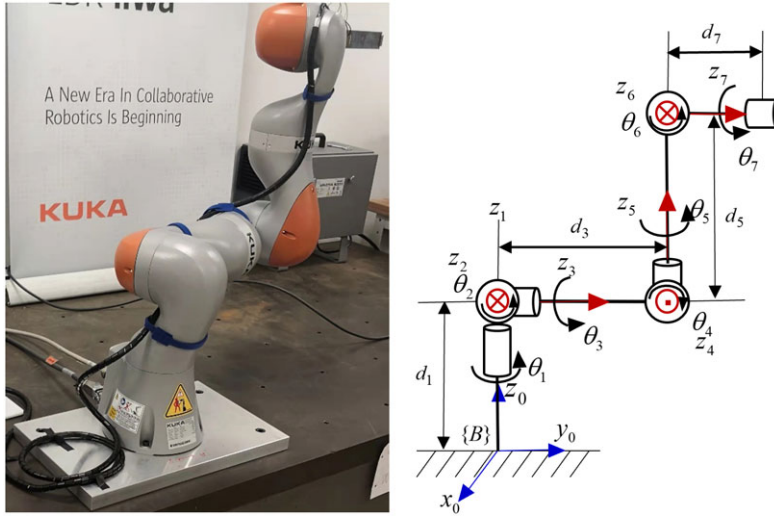
Figure 2. *Model diagram of the KUKA redundant robot with D−H parameters annotation.*

4. Simulation studies

In this phase, we firstly evaluate the performance of the proposed SFTMP method and then evaluate its fault-diagnosis performance. The validation is performed based on the KUKA redundant robot (LBR IIWA type, 7 revolute joints and 7 links). Its D−H parameters are shown in Table I and the model diagram of KUKA iiwa is shown in Fig. 2. The forward kinematics model is established based on homogeneous matrix chains with parameters configured in the D−H table. For the tracking tasks, the desired motion targets are circle and square (rectangular) paths. The radius of the circle path is 0.15 m, and the length of the square path is 0.20 m. The converging factor ϵ of the primary double network resolver for the developed system is set to 0.0001 and control gains of the fault-diagnosis algorithm (Algorithm 1) are set to $k = 1$, $k_1 = 10$ and $k_2 = 10$. The initial joint angle vector of the redundant robot is set as $\theta(0) = [\pi/12 \ \pi/4 \ 0 \ -\pi/2 \ 0 \ -\pi/12 \ \pi/9]^T$. The motion process lasts 30/40 s for the path tracking tasks.

4.1. Path tracking performance with locked/frozen joints

Fig. 3 shows the joint angles θ of the KUKA iiwa robot during its end-effector's path tracking process, and the desired paths are receptively a circle and a square. In the circle drawing task, as shown in Fig. 3(a), joint 3 will be locked with an immediate speed of $\dot{\theta}_3$ (after 5 s), joint 5 will be locked with 10 s, its velocity $\dot{\theta}_5$ will be set to 0, joint 7 will keep locked/frozen constantly leaving all the other to work
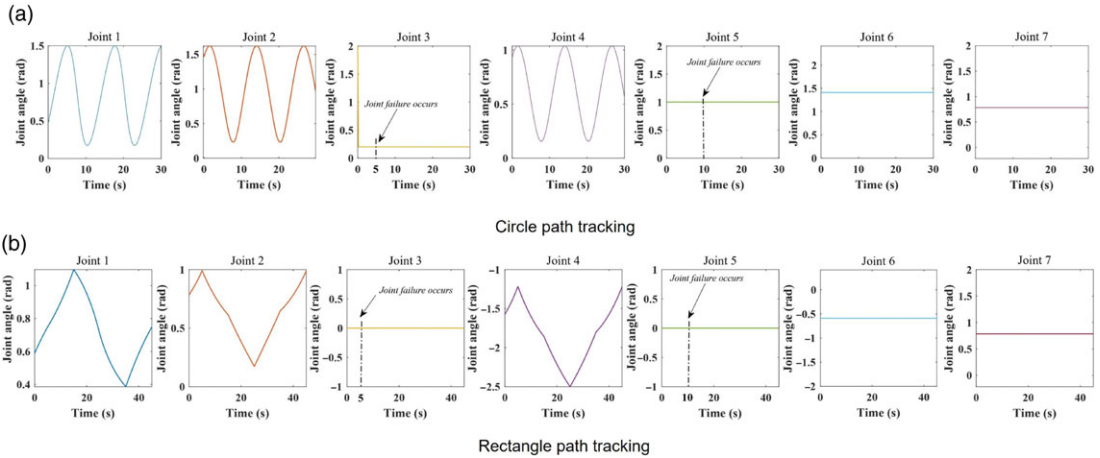
Figure 3. *Joint angular values θ of the KUKA iiwa robot while performing circle and square drawing tasks by SFTMP. Joint 3 and joint 5 are locked after 5 and 10 s, respectively. Joint 7 keeps locked/frozen in the entire motion process.*
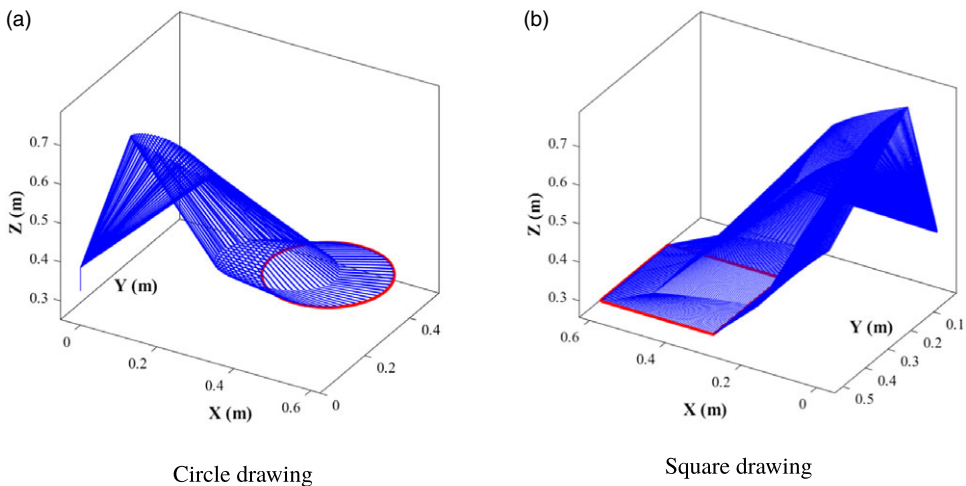

Figure 4. *Execution of the circle and square drawing tasks of the KUKA iiwa robot with joint locks by SFTMP.*

as normal, whose joint angles will be operated by the proposed sparse-based optimizer. In the tracing of the square path (Fig. 3(b)), joints 3 and 5 are locked after 5 and 10 s immediately with a 0 joint velocity $\dot{\theta}_3$ and $\dot{\theta}_5$, respectively. We keep joint 7 being locked/frozen constantly and all the other joints work as normal, which are controlled by our proposed method. Seen again from Fig. 3, only joints 1, 2, 4, and 6 work normally to fulfill the circle and square drawing tasks, and joint 6 is not actuated in the whole motion process but is not locked. Under these circumstances, Figs. 4 and 5 further, respectively, show the circle and square trajectories generation/execution tasks tested on the KUKA iiwa platform, driven by our proposed fault-tolerant manipulation method optimized by sparsity, with locks on joint 3 and joint 5 at their failure time instants, respectively. The piece-wise blue straight lines indicate the body of the connections with the redundant KUKA iiwa robot in Fig. 4(a) and (b) and the red lines depict the end-effector's motion trajectory created by the approach described in this paper. It can be observed that using the developed SFTMP, the end-effector of the KUKA iiwa robot can trace the appropriate circular and
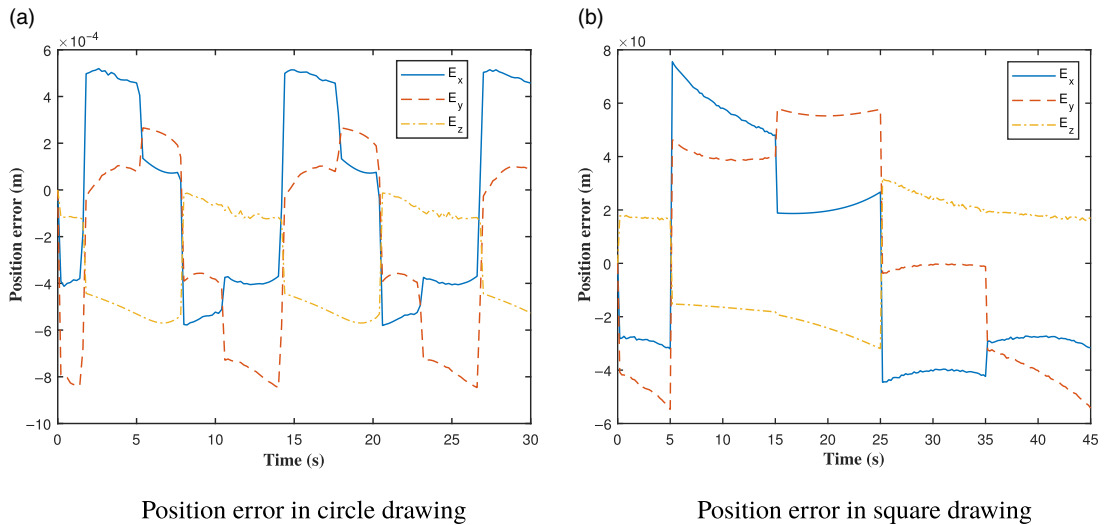
(a)

Position error in circle drawing

(b)

Position error in square drawing

Figure 5. *Position errors of the end-effector by SFTMP for circle and square trajectories.*

square trajectories with the optimized joint angles. Figure 5 further reveals the positioning failure of the end-effector of the KUKA iiwa robot in order to evaluate the accuracy of the proposed motion planner and the controller, and we can observe that the position errors are below 10×10^{-4} m in both scenarios. As observed again from Fig. 3, joint 6 is keeping the idle status during the motion control process, but it is not belonging to the joint failure cases, this might be owing to that the proposed sparse optimization-based redundancy resolution can produce sparse solutions for joint velocity $\dot{\theta}$. Moreover, the developed sparsity-based technique with joint lock can still ensure the path tracking accuracy as predicted, which is proved by the path tracking performance and the optimized joint angles of the KUKA iiwa robot. These all simulation results indicate the effectiveness of the proposed fault-tolerant motion planning mechanism for a redundant robot with an unanticipated joint lock/failure. In the meantime, sparse joint resolutions can be generated as well.

4.2. Fault diagnosis of joints

After validating SFTMP for the KUKA iiwa robot, further simulation studies on the recognition of locked joints by the proposed simultaneous fault diagnosis scheme are presented. Investigating again from the optimized joint angles by SFTMP in the previous subsection, we can see that joint 6 also "seems" to fall into a joint lock situation, but it is not a fault joint. To avoid the incorrect judgment of locked joints only from joint angle observations, we apply the proposed fault-diagnosis algorithm to distinguish which joints are real locked joints. Figure 6 shows the performable results of the proposed fault-diagnosis method (Algorithm 1) for locked joint recognition. The circle drawing case in Fig. 6(a) shows that $\tilde{\sigma}_3$, $\tilde{\sigma}_5$, and $\tilde{\sigma}_7$ cannot converge to zero in the steady state. Nevertheless, other values $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_4, \tilde{\sigma}_6$ in $\tilde{\Lambda}$ can converge to zero. Such phenomena of these values indicate that joints 3, 5, and 7 are fault joints, which is sealed by Fig. 3(a). Similarly, in the square drawing case from Fig. 6(b), $\tilde{\sigma}_3$, $\tilde{\sigma}_5$, and $\tilde{\sigma}_7$ cannot converge to zero finally and other values $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_4, \tilde{\sigma}_6$ in $\tilde{\Lambda}$ can converge to zero finally, which indicates that joints 3, 5, and 7 are fault joints, and this point is sealed by Fig. 3(b). All of these results demonstrate that the designed algorithm (Algorithm 1) is an efficient alternative to detect and locate the locked joints during the motion process and preserve the ability to avoid incorrect fault joint judgment. Therefore, the proposed sparsity-based method possesses the dual capability of fault-tolerant motion planning and simultaneous fault joint diagnosis.
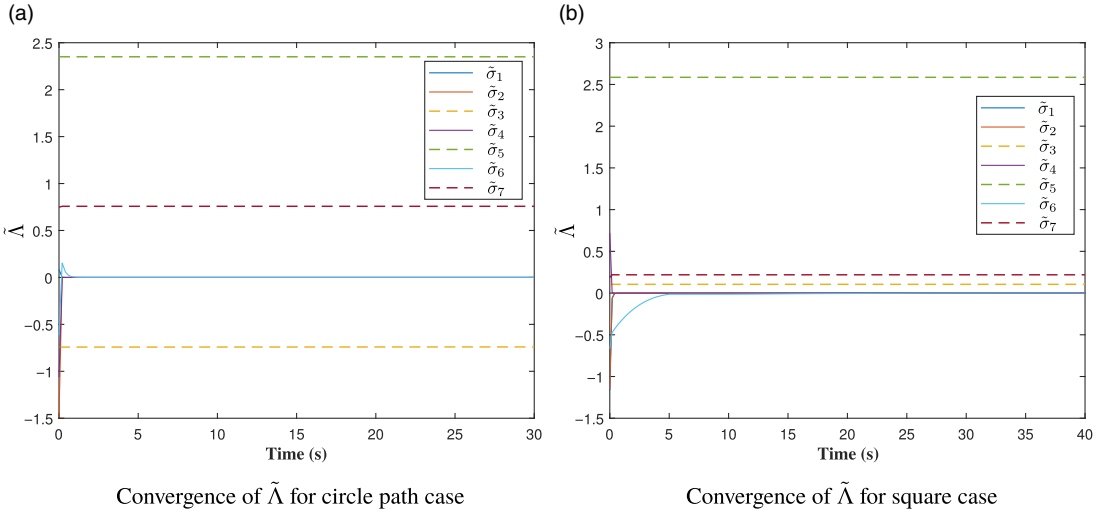
Figure 6. *Convergence of matrix $\tilde{\Lambda}$. Elements $\tilde{\sigma}_3$, $\tilde{\sigma}_5$, and $\tilde{\sigma}_7$ cannot converge to zero finally, indicating that the corresponding joints 3, 5, and 7 are locked joints. Other elements $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_4, \tilde{\sigma}_6$ converge to zero, indicating that the corresponding joints 1, 2, 4, and 6 are normal working joints.*

4.3. Sparsity comparison

Figure 7 shows the sparsity comparison results between the proposed sparsity-based method and the original method without sparsity formulation [18] for the same aforementioned path tracking tasks. Evidently, we could observe that, both in the circle and square drawing cases, the proposed method presents better sparsity as it shows lower magnitude values of the sparsity index. Table II further shows the quantitative comparisons of the average sparsity among the computed sparsity with different index parameters $p = 0.4, 0.6, 0.8, 1, 1.5, 2$. Specifically, for the circle drawing, the average sparsity by the proposed method is -3.6811 ± 1.4823 which is lower than the average sparsity -6.5591 ± 4.7162 by the original method; for the square drawing, the average sparsity by the proposed method is -1.6614 ± 0.5468 which is lower than the average sparsity -2.6172 ± 3.9801 by the original method. We see that lower average amplitudes of sparsity values can be achieved by the proposed method as compared with the original method [18], and the sparsity of joint actuation by the proposed method can be enhanced by around 43.87% and 36.51%, respectively, for tracking tasks of circle and square paths. All of these results reveal that, under the same types of path tracking targets, not only the proposed method can fulfill the fault-tolerant motion planning and control, but also it can enhance the sparsity performance of the redundant robot system and decrease unnecessary joint liveness.

5. Experiment results

In this section, a further experimental study on the real robot platform of the KUKA iiwa has been conducted, after verifying it from the aspect of simulation. As a circular route with a radius of 0.15 m the intended route for motion planning and control is established, the aforementioned sparsity-based technique utilized to summarize the newly proposed primary dual neural network resolver to achieve the ideal joint angles for the KUKA iiwa robot. A pen is fitted to draw the circle and the working period for data capturing is restricted to 300 sampling frames.

Figure 8 presents the experimental snapshots of executing the desired circle drawing task of the KUKA iiwa robot. It can be seen that the end-effector of the KUKA iiwa robot successfully accomplishes the designed circle trajectory to finish drawing for recordings. In Fig. 8, the final recorded trajectory

Table II. *Average sparsity of joints of the redundant robot system during motion planning and control.*
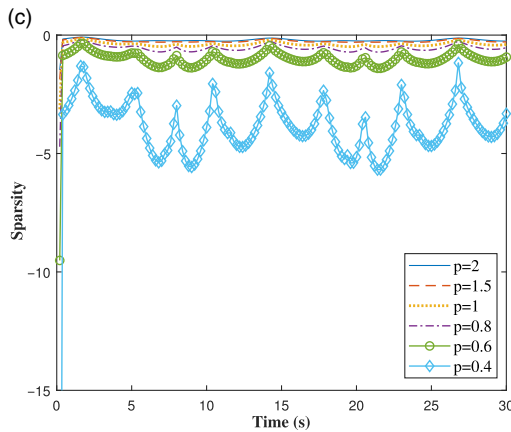
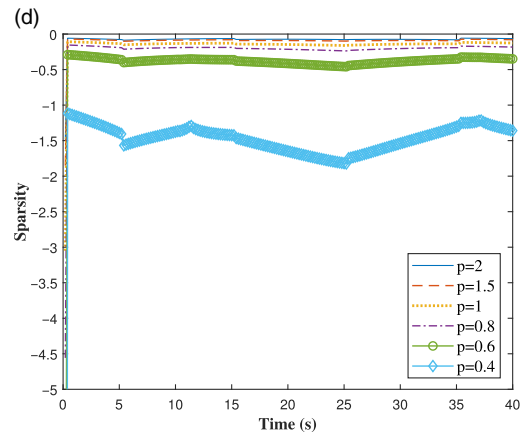| Desired path	The proposed method	Without sparsity formulation
Circle path	-3.6811 ± 1.4823	-6.5591 ± 4.7162
Square path	-1.6614 ± 0.5468	-2.6172 ± 3.9801
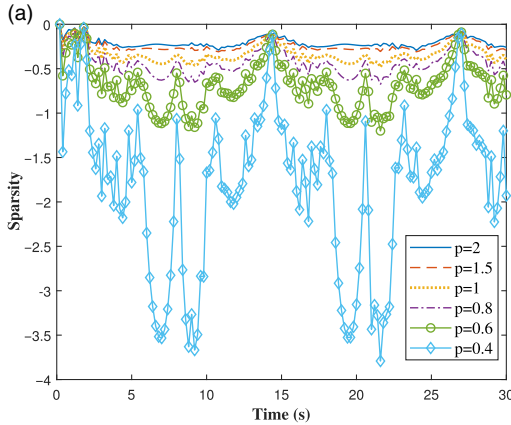
(a)
Sparsity for circle path case by proposed method
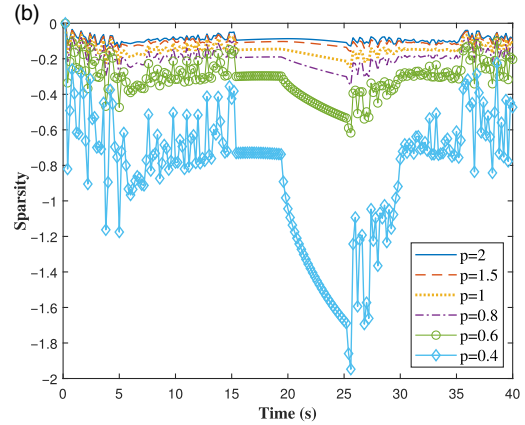
(b)
Sparsity for square path case by proposed method

(c)
Sparsity for circle path case by method [18]

(d)
Sparsity for square path case by method [18]

Figure 7. *Sparsity index comparison between the proposed sparsity-based method and the original method [18]. The sparsity indices $-\|\dot{\theta}\|_p$ (amplitude values) of the proposed method are generally lower than those of method [18].*

of the KUKA iiwa robot by SFTMP is shown, and it can be observed that the robot well tracks the circle path eventually. Furthermore, from Fig. 9 we can see, joint failure occurs in joints 3 and 5 at particular times during the KUKA iiwa robot motion execution with its velocity reaching 0. In particular, after approximately 100 sampling frames, joint 3 falls into the lock condition and joint 5 keeps locked at the start of the motion process. Table III shows the positions of 12 sampled points in the actual path and 12 sampled points the desired path, respectively, and it can be seen that the position errors

Table III. *Twelve samples of the coordinates of the designed and actual trajectories, respectively.*

| Point #	(X_d, Y_d) (mm)	(X, Y) (mm)	Position error (m)
1	(448.2, 471.9)	(447.6, 472.7)	$(0.6, 0.8) \times 10^{-3}$
2	(610.9, 447.3)	(610.4, 446.4)	$(0.5, 0.9) \times 10^{-3}$
3	(656.9, 365.3)	(655.8, 364.3)	$(0.9, 1.0) \times 10^{-3}$
4	(659.7, 325.8)	(658.6, 325.6)	$(1.1, 0.2) \times 10^{-3}$
5	(580.2, 202.4)	(579.1, 203.4)	$(1.1, 1.0) \times 10^{-3}$
6	(537.2, 187.4)	(536.8, 188.5)	$(0.4, 0.9) \times 10^{-3}$
7	(450.8, 197.1)	(451.0, 197.5)	$(0.8, 0.4) \times 10^{-3}$
8	(398.5.,435.3)	(397.7, 435.8)	$(0.8, 0.5) \times 10^{-3}$
9	(505.0, 484.9)	(505.7, 486.3)	$(0.7, 1.4) \times 10^{-3}$
10	(547.6, 480.1)	(547.8, 481.6)	$(0.2, 0.5) \times 10^{-3}$
11	(374.9, 269.7)	(375.1, 270.0)	$(0.2, 0.3) \times 10^{-3}$
12	(361.6, 357.3)	(361.2, 357.7)	$(0.4, 0.4) \times 10^{-3}$
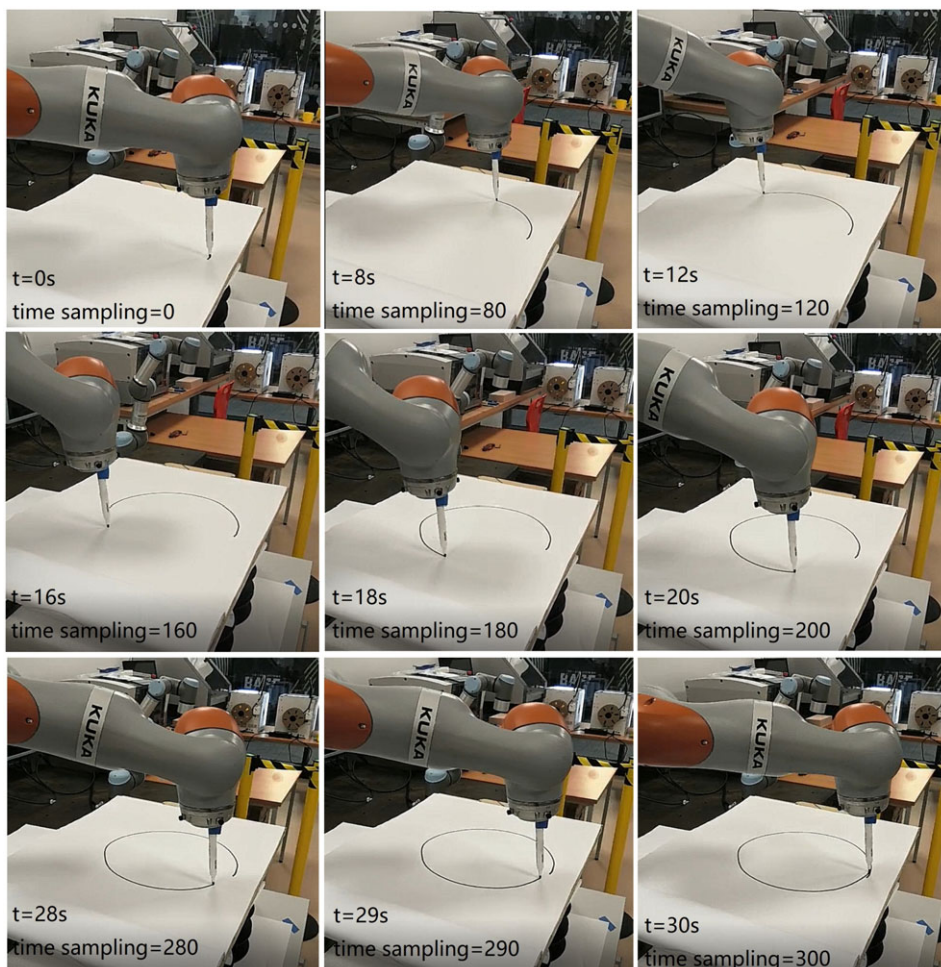
Figure 8. *Snapshots from experiments of the KUKA iiwa robot while drawing a circle path with locked joints based on SFTMP.*
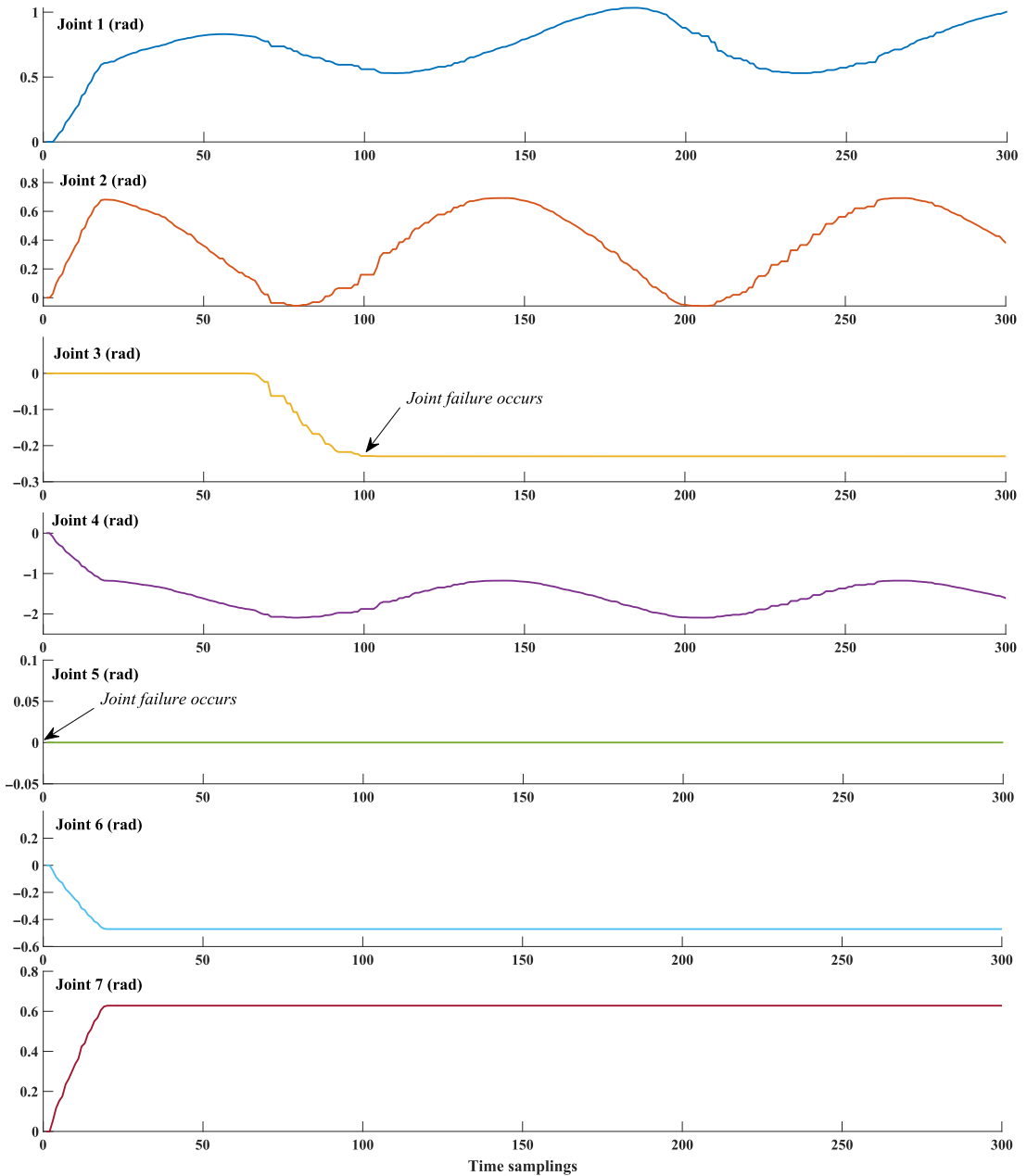
Figure 9. *Illustration of the actual joint angles of the KUKA iiwa robot. Joint 5 keeps locked throughout the entire motion process and joint 3 comes into being locked after around 100 sampling frames.*

$(|X_d - X|, |Y_d - Y|)$ can be within 1.5×10^{-3} m which is rather small as compared with the radius of the circle path to track. We may deduce that, in conjunction with path tracks as illustrated in Fig. 8, with two joints being trapped into the lock states at some unexpected time instants, the KUKA iiwa robot can still keep to let its end-effector track the path well based on SFTMP.

The effectiveness of the proposed algorithm has been well validated by both the simulation and experimental studies. From the simulation result, the simulated KUKA iiwa robot was able to execute the circle and rectangle drawing tasks under the circumstance that two of the seven joints are at fault; the

experiment was also conducted on the real-world robot platform and the KUKA iiwa could also complete the drawing tasks well with fault tolerance, which would not be feasible if the proposed algorithm was not built in the controller of the KUKA iiwa robot. The employment of SFTMP seeks the help of the redundancy where not all the joints are involved in while performing certain tasks. To draw a conclusion, the efficacy of the developed sparse technique for the fault-tolerant function-driven manipulation of redundant robots is demonstrated by all these experimental findings.

6. Conclusion

In this work, in order to manage potential joint lock situations frequently encountered in redundant robots in ordinary industrial operations, a new fault-tolerant motion planning method in a sparsity perspective is proposed for motion control of redundant robots, and a simultaneous fault-diagnosis paradigm is developed with the proposed sparsity-based method to adaptively detect the lock joints. The promising completion with smoothness of the circle and square drawing tasks by SFTMP can be archived under joint lock situations happen. Simulation and experimental studies on the KUKA iiwa robot platform show SFTMP's effectiveness on the joint locking problems of redundant robots. The future works will be focused on the effectiveness of the method along other random paths than circle and square path and the implementation of SFTMP into real-time control of a robot.

Competing interest declaration. The authors declare none completing interest.

Author contributions. ZL designed the algorithm; CL built the algorithm into the robot controller. ZL and CL prototyped the whole system and wrote the paper. SZ was the pilot for the tests and dealt with the hardware integration. ZL, CL, SZ, and HS carried out the software integration and helped with the tests. ZL and SL conceived the design idea and supervised the project. CL led the project reviewed work progress, and the paper revision.

Supplementary material. To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574722000285.

References

[1] X. Yu, W. He, Q. Li, Y. Li and B. Li, "Human-robot co-carrying using visual and force sensing," *IEEE Trans. Ind. Electron.* **68**(9), 8657–8666 (2021).

[2] Z. Mu, B. Zhang, W. Xu, B. Li and B. Liang, "Fault Tolerance Kinematics and Trajectory Planning of a 6-DOF Space Manipulator Under a Single Joint Failure," **In:** *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)* (IEEE, 2016) pp. 483–488.

[3] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robotic. Autom.* **13**(4), 622–629 (1997).

[4] K. Li and Y. Zhang, "Fault-tolerant motion planning and control of redundant manipulator," *Control Eng. Pract.* **20**(3), 282–292 (2012).

[5] A.-N. Sharkawy, P. N. Koustoumpardis and N. Aspragathos, "A recurrent neural network for variable admittance control in human–robot cooperation: Simultaneously and online adjustment of the virtual damping and inertia parameters," *Int. J. Intell. Robot. Appl.* **4**(4), 441–464 (2020).

[6] X. Yu, W. He, H. Li and J. Sun, "Adaptive fuzzy full-state and output-feedback control for uncertain robots with output constraint," *IEEE Trans. Syst. Man Cybern. Syst.* **51**(11), 6994–7007 (2021).

[7] X. Yu, B. Li, W. He, Y. Feng, L. Cheng and C. Silvestre, "Adaptive-constrained impedance control for human-robot co-transportation," *IEEE Trans. Cybern.*, 1–13 (2021).

[8] P. Chen, J. Xiang and W. Wei, "A unified weighted least norm method for redundant manipulator control," *Int. J. Adv. Robot. Syst.* **13**(19), 1–10 (2016).

[9] 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 Trans. Neural Netw.* **14**(3), 658–667 (2003).

[10] 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 Trans. Mechatron.* **22**(3), 1423–1434 (2017).

[11] L. Jin, S. Li, X. Luo, Y. Li and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018).

[12] Y. Chen, J. E. McInroy and Y. Yi, "Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance," *IEEE Trans. Robotic. Autom.* **19**(4), 680–691 (2003).

[13] R. G. Roberts, H. G. Yu and A. A. Maciejewski, "Fundamental limitations on designing optimally fault-tolerant redundant manipulators," *IEEE Trans. Robot.* **24**(5), 1224–1237 (2008).

[14] 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 Trans. Robot.* **30**(2), 516–524 (2014).

[15] B. Xie and A. A. Maciejewski, "Kinematic design of optimally fault tolerant robots for different joint failure probabilities," *IEEE Robot. Autom. Lett.* **3**(2), 827–834 (2018).

[16] K. Li and Y. Zhang, "Fault-tolerant motion planning and control of redundant manipulator," *Control Eng. Pract.* **20**(3), 282–292 (2012).

[17] 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)* (2018) pp. 533–538.

[18] Z. Li, C. Li, S. Li and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Trans. Ind. Inform.* **16**(12), 7469–7478 (2020).

[19] Z. Gao, C. Cecati and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.* **62**(6), 3757–3767 (2015).

[20] M. A. Costa, B. Wullt, M. Norrlöf and S. Gunnarsson, "Failure detection in robotic arms using statistical modeling, machine learning and hybrid gradient boosting," *Measurement* **146**(7), 425–436 (2019).

[21] F. Bach, R. Jenatton, J. Mairal and G. Obozinski, "Convex optimization with sparsity-inducing norms," *Optim. Mach. Learn.* **5**, 19–53 (2011).

[22] J. Xu and J. Wang, "Effective motion planning of manipulator based on SDPS-RRTConnect," *Robotica*, 1–13 (2021).

[23] R. Polvara, S. Sharma, J. Wan, A. Manning and R. Sutton, "Autonomous vehicular landings on the deck of an unmanned surface vehicle using deep reinforcement learning," *Robotica* **37**(11), 1867–1882 (2019).

[24] L. Bascetta, G. Ferretti and B. Scaglioni, "Closed form Newton–Euler dynamic model of flexible manipulators," *Robotica* **35**(5), 1006–1030 (2017).

[25] M. S. Asif and J. Romberg, "Fast and accurate algorithms for re-weighted -norm minimization," *IEEE Trans. Signal Process* **61**(23), 5905–5916 (2013).

[26] I. Selesnick, "Sparse regularization via convex analysis," *IEEE Trans. Signal Process.* **65**(17), 4481–4494 (2017).

[27] S. Boyd and L. Vandenberghe. *Convex Optimization* (Cambridge University Press, 2004).

[28] X.-B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Trans. Neural Netw.* **14**(2), 426–431 (2003).

[29] X.-B. Gao and L.-Z. Liao, "A neural network for monotone variational inequalities with linear constraints," *Phys. Lett. A* **307**(2), 118–128 (2003).