IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 August 2022, accepted 5 September 2022, date of publication 12 September 2022, date of current version 21 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206365

RESEARCH ARTICLE

A Provably Constrained Neural Control Architecture With Prescribed Performance for Fault-Tolerant Redundant Manipulators

RAJPAL SINGH AND JISHNU KESHAVAN, (Member, IEEE)
Department of Mechanical Engineering, Indian Institute of Science, Bengaluru, Karnataka 560012, India
Corresponding author: Rajpal Singh (rajpalsingh@iisc.ac.in)

This work was supported in part by the Institute of Eminence under Grant IE/RERE-21-0537.08.

ABSTRACT In this paper, a novel neural control architecture is proposed and investigated for resolving redundancy in trajectory tracking applications for manipulators with joint velocity constraints. First, a nonlinear invertible map is invoked to transform the constrained system state into a set of unconstrained variables, which allows the proposed framework to realize solutions that rigorously adhere to the specified bound constraints. Next, a quadratic program (QP) architecture is synthesized by incorporating suitably prescribed performance constraints to ensure that the resulting system error achieves exponential convergence to the ground truth while also ensuring that the system states evolve along trajectories with good transient and steady-state behavior. Thus, in contrast with previous approaches that do not rigorously guarantee the satisfaction of the bound constraints in the transient phase and/or the steady-state, the proposed scheme ensures that these constraints are rigorously satisfied while achieving prescribed performance both during the transient phase and in the steady-state. The novelty of the proposed scheme lies in the fusion of prescribed performance constraints with the state and input constraints within the QP framework, which offers the important advantage of higher computational efficiency compared to leading alternative designs. A detailed theoretical analysis is undertaken to prove the global stability and convergence of the proposed scheme. Simulation and experimental results with the KUKA LBR IIWA 14 R820 manipulator are used to verify the efficacy of the proposed scheme in accomplishing trajectory tracking for the fault-free and fault-tolerant cases with multiple joint failures. Finally, detailed performance comparison studies with leading alternative designs are further used to illustrate the advantages of the proposed scheme.

INDEX TERMS Fault tolerance, joint constraints, prescribed performance, redundancy resolution, zeroing neural networks.

I. INTRODUCTION

A redundant robotic manipulator refers to a manipulator which possesses more degrees of freedom than it is required to complete a task objective. The presence of these additional degrees of freedom leads to multiple sets of solutions for the same problem statement, thus allowing for the enforcement of additional sets of constraints on the manipulator in addition to the primary task. A fundamental problem in robotics is the problem of redundancy resolution for trajectory tracking applications using redundant manipulators. It refers to the computation of the joint poses required to achieve desired end-effector path in the Cartesian space. Due to the nonlinear nature of the mapping between the joint pose and end-effector positions, redundancy resolution via inverse kinematics is often quite difficult. Hence, redundancy resolution is usually performed at the velocity level, which can be represented in the form of a time-varying underdetermined linear equation [1], [2].

The zeroing neural network (ZNN) represents a special class of recurrent neural networks that is primarily aimed at finding the zeros of a time-varying system represented by a set of linear or nonlinear equations. ZNNs were first proposed for finding the solution of the time-varying Sylvester's

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

equation [3]. Compared to the traditional gradient-based neural networks (GNNs), these neural network architectures are better suited for finding the solution of time-varying problems related to trajectory tracking, which is attributed to the fact that the time-derivative information of the time-varying coefficients is incorporated seamlessly within the ZNN architecture, thus providing a stark contrast with the traditional GNN framework [4]. As such, numerous ZNN architectures have been proposed for finding the solution of various time-varying problems of practical interest [5], [6], [7], [8], [9], [10]. In particular, ZNN models are proposed as solutions to the dynamic matrix inversion problem in [5], [6], and [7], time-varying linear inequalities in [8] and [9], and time-varying nonlinear inequalities in [10].

Over the years, a few prominent studies have been conducted for finding the solution of time-varying underdetermined linear systems with application to redundancy resolution. Two new ZNN models are proposed in [11] for this purpose, while a gain-adjustment neural network (GANN) that ensures super-exponential convergence to the ground truth is proposed in [12] for finding the solution of an underdetermined linear system. Further, a projected ZNN (PZNN) based joint drift-free scheme is proposed in [13] for redundancy resolution of a manipulator in the presence of noise. However, the previously mentioned studies do not include constraints on joint velocity. In contrast, a novel ZNN model is proposed in [14] that relies on equality transformations for finding the online solution of a time-varying system of linear equations with bound constraints. Subsequently, an equivalent discrete-time version of this ZNN model is proposed in [15], and a noise-tolerant ZNN for the solution of a time-varying underdetermined linear system of equations with bounds on both the state and residual error is proposed in [16]. Further, a ZNN model with finite-time convergence is proposed in [17] and a ZNN model based on infinity norm minimization is proposed in [18] for the solution of an underdetermined system with state and input constraints. However, these studies do not always guarantee rigorous enforcement of the state constraints either during the transient phase or in steady-state, which is a critical drawback, especially for redundancy resolution applications in robotics.

To overcome this drawback, a novel quadratic program (QP)-based neural control architecture is proposed in this paper, which incorporates prescribed performance constraints with state and input constraints at the joint velocity level. First, the joint velocity constraints are implemented via an invertible nonlinear map to transform the constrained system state into a new set of unconstrained variables. This transformation ensures that the bound constraints are rigorously adhered to during the solution process, distinguishing it from the approaches proposed in [14], [15], and [17] that provide no such formal guarantees, especially in the transient state. The proposed model further incorporates suitably prescribed performance constraints, which guarantee desired transient and steady-state performance characteristics, and have been shown to deliver prescribed performance for various dynamical systems [19], [20], [21], [22]. Prescribed performance control ensures that the system forces the tracking error to converge to a small residual set with a convergence rate greater than or equal to some predefined value in such a manner that the overshoot does not exceed a predefined limit. The proposed model uses a convex optimization framework to enforce both the bounds on the state and the input, and the prescribed performance constraints simultaneously. Combining these approaches and incorporating them within a quadratic program further ensures that the solutions obtained from the proposed model are optimal in accomplishing the task objective. In particular, the optimization framework is designed with the aim of enforcing bounds on the mathematical range of the solution, which is in contrast with the ZNN-based optimization model proposed in [23], which does not impose any bounds on the system state/input. Moreover, the proposed framework is used to impose explicit exponentially convergent dynamic bounds on the residual error as opposed to the study in [16], where only static bounds are imposed on the tracking error and thus cannot guarantee desired transient and steady-state performance. Theoretical analysis is undertaken to demonstrate the stability and convergence properties of the proposed scheme. Complexity analysis is also undertaken to demonstrate the computational efficiency of the proposed model as compared to leading alternative designs such as [14], [24]. Simulation and experimental studies are carried out with a KUKA LBR IIWA 14 R820 to show the efficacy of the proposed models for path tracking applications in redundant manipulators with bounds on the joint velocities. The suitability of the proposed model for fault-tolerant trajectory tracking is then demonstrated in simulation and experiments in the presence of multiple joint failures. Note that this approach differs from the previous approaches proposed for fault tolerance such as [24], in the sense that it incorporates limits on the state/input of the system. Further, the presence of prescribed performance constraints and the inclusion of formal guarantees for rigorous adhesion to bound constraints distinguishes it from the fault-tolerant approaches proposed in [25], [26], and [27]. To the best of the authors' knowledge, no other study has focused on the constrained resolution of redundant manipulators that delivers prescribed transient and steady-state performance of the tracking error within an optimal framework.

The main contributions of this paper are summarized below.

i) A novel QP-based neural control framework is introduced for redundancy resolution with bound constraints at the joint velocity level. To the best of the authors' knowledge, this is the only framework that incorporates both prescribed performance and joint velocity constraints within an optimization framework for redundancy resolution in fault-tolerant trajectory tracking applications.

ii) The proposed model relies on an invertible nonlinear map to transform the constrained system state into a new set of unconstrained variables. This transformation formally guarantees that the proposed model will rigorously satisfy the

joint constraints both during the transient phase and in the steady-state, which is in contrast with most previous studies.

iii) Further, the proposed model also incorporates prescribed performance constraints which allow the proposed model to exert control over the transient and steady-state behavior of the system error, thus leading to superior performance in contrast with previous studies.

iv) For robotic trajectory tracking applications, the adoption of the proposed model is shown to deliver superior trajectory tracking performance in comparison with leading alternative designs both in the presence and absence of multiple joint failures.

The rest of the paper is organized as follows. Section II describes the problem formulation for redundancy resolution with bound constraints at the joint velocity level. Section III presents the analytical formulation of the proposed model with the associated theoretical analysis. The application of the proposed ZNN models to KUKA LBR IIWA 14 R820 for trajectory tracking in the presence and absence of joint failures is shown in Section IV. A detailed performance comparison study with leading alternative designs is also included. The conclusions of the paper are presented in Section V.

II. PROBLEM FORMULATION

In this section, the problem of redundancy resolution for systems involving inverse kinematics with bound constraints on the joint velocity is introduced. The problem of redundancy resolution can be stated as follows. Given a desired path $\boldsymbol{r_d} \in \mathbb{R}^m$ of the end-effector in the task space, the required joint angles $\boldsymbol{\theta} \in \mathbb{R}^n$ in the joint space have to be determined. The solution to this problem can be achieved by solving the forward kinematics of the manipulator, which is given as

$$\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{r_d}, \tag{1}$$

where $\boldsymbol{f}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ represents the forward kinematic mapping. Due to the nonlinear nature of the mapping, it is very difficult to solve (1) by finding the inverse mapping for $\boldsymbol{f}(\cdot)$. Hence, this problem is usually solved at the velocity level. Redundancy resolution at the velocity level [28], [29], [30] with bound constraints is achieved by the solution of the time-dependent underdetermined system with the bounds on the joint velocity given below

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

$$\text{subject to } \dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}}(t) \leq \dot{\boldsymbol{\theta}}^+, \tag{3}$$

where $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{m\times n}$ represents the Jacobian matrix, $\dot{\boldsymbol{\theta}}(t) \in \mathbb{R}^{n\times 1}$ represents the joint velocity, and $\dot{\boldsymbol{r}}_d \in \mathbb{R}^{m\times 1}$ represents the desired end-effector velocity. $\dot{\boldsymbol{\theta}}^-$ and $\dot{\boldsymbol{\theta}}^+$ represent the constant limits on joint velocity.

Remark 1: This study considers only the bounds at the velocity level. However, the problem definition can be easily extended to include limits on the joint angular position level by making use of dynamic constraints as $\boldsymbol{\xi}^-(t) \leq \dot{\boldsymbol{\theta}}(t) \leq \boldsymbol{\xi}^+(t)$ [31], where the individual components of the dynamic bounds are defined as $\xi_i^-(t) = \max\{\kappa_p(\theta_i^- - \theta_i(t)), \dot{\theta}_i^-\}$, $\xi_i^+(t) = \min\{\kappa_p(\theta_i^+ - \theta_i(t)), \dot{\theta}_i^+\}$, where θ_i^- and θ_i^+ represent the limits on the angular position of the i^{th} joint, and $\kappa_p > 0$ is a scaling factor for the joint limits.

III. QP-BASED NEURAL CONTROL ARCHITECTURE

In this section, a quadratic program-based ZNN embedded with performance constraints is proposed for the solution of system (2) subject to bound constraints (3) along with the subsequent theoretical analysis to demonstrate the convergence properties of the proposed scheme. For the proposed scheme, both the system constraints and the prescribed performance constraints are included in the formulation via a nonlinear mapping, and a QP framework is subsequently introduced to impose both of them simultaneously.

A. CONSTRAINED VELOCITY AND ERROR TRANSFORMATION

In this subsection, a nonlinear transformation is introduced to impose the constraints on the state of the system. Subsequently, prescribed performance constraints (PPCs) are incorporated into a ZNN framework for better transient and steady-state performance.

In order to ensure that the state of the system remains within the given bounds $\forall\, t \in [0, \infty)$, a nonlinear transformation that converts the system state $\dot{\boldsymbol{\theta}}(t)$ to an unconstrained variable $\boldsymbol{\chi}(t)$ is given below [21]

$$\eta_i = \frac{\dot{\theta}_i(t) - \dot{\theta}_i^-}{\dot{\theta}_i^+ - \dot{\theta}_i^-} = \frac{e^{\chi_i}}{1 + e^{\chi_i}} \qquad i = 1, 2, \ldots, n. \tag{4}$$

where $\eta_i \;\forall\; i = 1, 2, .., n$ represents the required nonlinear transformation. Then, by introducing the variables, $\boldsymbol{\Gamma} = [\Gamma_1\ \Gamma_2\ \ldots\ \Gamma_n]^\top \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{\chi} = [\chi_1\ \chi_2\ \ldots\ \chi_n]^\top \in \mathbb{R}^{n\times 1}$, the components of the transformed variable vector $\boldsymbol{\chi}(t)$ can be obtained from (4) as

$$\chi_i(t) = ln\left(\frac{\eta_i}{1-\eta_i}\right) = ln\left(\frac{\dot{\theta}_i(t) - \dot{\theta}_i^-}{\dot{\theta}_i^+ - \dot{\theta}_i(t)}\right) = \Gamma_i(\dot{\theta}_i). \tag{5}$$

where $\boldsymbol{\Gamma}(\dot{\boldsymbol{\theta}}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ represents the mapping from the constrained variable $\dot{\boldsymbol{\theta}}(t)$ to the unconstrained variable $\boldsymbol{\chi}(t)$. It is straightforward to observe that the transformation $\boldsymbol{\Gamma}(\dot{\boldsymbol{\theta}})$ is a smooth monotonically increasing function, and hence is invertible. Consequently, it is evident that while $\boldsymbol{\chi}(t)$ remains bounded within the limits $(-\infty, \infty)$, the state $\dot{\boldsymbol{\theta}}(t)$ remains bounded between the bounds $(\dot{\boldsymbol{\theta}}^-, \dot{\boldsymbol{\theta}}^+)$, fulfilling the constraints imposed by (3). Thus, the objective of finding the solution of (2) subject to (3) now translates to driving the unconstrained variable $\boldsymbol{\chi}(t)$ to its desired value.

Introducing the tracking error $\boldsymbol{\epsilon}(t) = \boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{r_d}$, and using a non-negative constant $k \geq 0$, the system error is defined as

$$\begin{aligned} \boldsymbol{e}(t) &= \dot{\boldsymbol{\epsilon}}(t) + k\boldsymbol{\epsilon}(t) \\ &= \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) - \dot{\boldsymbol{r}}_d + k(\boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{r_d}) \\ &= \boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}) - \dot{\boldsymbol{r}}_d + k\,(\boldsymbol{f}\,(\boldsymbol{\theta}) - \boldsymbol{r_d})\,. \end{aligned} \tag{6}$$

where $\boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ represents the inverse mapping for the transformation $\boldsymbol{\Gamma}(\dot{\boldsymbol{\theta}})$. To improve tracking perfor-

mance, especially in the presence of multiple joint failures, we consider the inclusion of the performance constraints to impose bounds on the system error (6) through the following transformation [21]

$$\tilde{\eta}_i = \frac{e_i(t) - \rho_l}{\rho_r - \rho_l} = \frac{e^{\tilde{\chi}_i}}{1 + e^{\tilde{\chi}_i}} \quad i = 1, 2, \ldots, m, \tag{7}$$

where $\tilde{\eta}_i \ \forall \ i = 1, 2, .., m$ represents the required nonlinear transformation. $\rho_r = \rho_0 e^{-\delta t} + \rho_\infty$ and $\rho_l = -\rho_r$ represent the time-varying prescribed performance constraints (PPCs) on the error function (6). Here, ρ_0 and ρ_∞ represent the initial and steady-state values of the performance constraints respectively. A straightforward choice for the parameter ρ_0 is $\rho_0 = ||\boldsymbol{e}(0)|| + \kappa_0$ for some user-defined constant κ_0, and $\rho_\infty \ll \rho_0$. The constant δ scales the convergence rate of the performance constraints and hence, allows for control over the system's transient performance.

Similar to (5), we introduce the vectors $\tilde{\boldsymbol{\chi}} = [\tilde{\chi}_1 \ \tilde{\chi}_2 \ \ldots \ \tilde{\chi}_m]^\top \in \mathbb{R}^{m\times 1}$ and $\tilde{\boldsymbol{\Gamma}} = [\tilde{\Gamma}_1 \ \tilde{\Gamma}_2 \ \ldots \ \tilde{\Gamma}_m]^\top \in \mathbb{R}^{m\times 1}$, and consider the component-wise transformation by using (7) as

$$\tilde{\chi}_i(t) = ln\left(\frac{\tilde{\eta}_i}{1 - \tilde{\eta}_i}\right) = ln\left(\frac{e_i(t) - \rho_l}{\rho_r - e_i(t)}\right) = \tilde{\Gamma}_i(e_i). \tag{8}$$

where $\tilde{\boldsymbol{\Gamma}}(\boldsymbol{e}) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ represents the mapping from the constrained variable $(\boldsymbol{e}(t))$ to the unconstrained variable $(\tilde{\boldsymbol{\chi}}(t))$. Again, owing to the invertible map $\tilde{\boldsymbol{\Gamma}}(\tilde{\boldsymbol{\chi}})$, it is apparent that the component-wise error function $e_i(t)$ remains bounded between the bounds $(\rho_l, \ \rho_r)$ provided that $\tilde{\chi}_i$ remains uniformly bounded such that $\tilde{\chi}_i \in (-\infty, \ \infty) \ \forall 1 \leq i \leq m$. Hence, the rigorous satisfaction of the performance constraints is ensured.

B. QP FRAMEWORK WITH JOINT VELOCITY AND PERFORMANCE CONSTRAINTS

In this subsection, a QP-based neural control architecture is developed by combining both the joint velocity constraints and the prescribed performance constraints. Further, a comparison study of the computational efficiency is undertaken relative to the schemes proposed in [14] and [24].

To account for the bounds on the error function, a vector $\boldsymbol{g}(\tilde{\boldsymbol{\chi}})$ is defined as

$$\boldsymbol{g}(\tilde{\boldsymbol{\chi}}) = \frac{1}{2}\tilde{\boldsymbol{\chi}} \circ \tilde{\boldsymbol{\chi}} = \frac{1}{2}\left[\tilde{\chi}_1^2 \ \tilde{\chi}_2^2 \ \cdots \ \tilde{\chi}_m^2\right]^\top \in \mathbb{R}^{m\times 1}, \tag{9}$$

where $\circ$ represents the Hadamard product. To satisfy both the velocity constraints and the performance constraints simultaneously, a convex optimization program is formulated by invoking the invertible transformation (5) as

$$\arg\min_{\dot{\boldsymbol{\theta}}(t)} \ \frac{1}{2}\boldsymbol{\chi}(\dot{\boldsymbol{\theta}})^\top \boldsymbol{\chi}(\dot{\boldsymbol{\theta}}) \text{ subject to } \boldsymbol{g}(\tilde{\boldsymbol{\chi}}(\dot{\boldsymbol{\theta}})) = \boldsymbol{0}. \tag{10}$$

To solve this optimization problem, the Lagrangian function is expressed as [32]

$$L = \frac{1}{2}\boldsymbol{\chi}^\top \boldsymbol{\chi} + \boldsymbol{\lambda}^\top \boldsymbol{g}(\tilde{\boldsymbol{\chi}}), \tag{11}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \ \ldots \ \lambda_m]^\top \in \mathbb{R}^{m\times 1}$ represents the Lagrange's multipliers. The design methodology used in [23] is adopted to formulate the desired ZNN model. The augmented state vector for the system is given as

$$\boldsymbol{y}(t) = \left[\dot{\boldsymbol{\theta}}(t)^\top, \ \boldsymbol{\lambda}^\top\right]^\top \in \mathbb{R}^{(m+n)\times 1}. \tag{12}$$

A vector $\boldsymbol{h}(\boldsymbol{y}(t), t)$ is then defined as

$$\boldsymbol{h}(\boldsymbol{y}(t), t) = \left(\frac{\partial L}{\partial \boldsymbol{y}}\right)^\top = \left[\left(\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}}\right) \ \frac{\partial L}{\partial \boldsymbol{\lambda}}\right]^\top \in \mathbb{R}^{(m+n)\times 1}. \tag{13}$$

For the solution of problem (10), a ZNN model is formulated to drive the components of $\boldsymbol{h}(\boldsymbol{y}(t), t)$ to zero as

$$\dot{\boldsymbol{h}}(\boldsymbol{y}(t), t)) = -\gamma \boldsymbol{\Phi}(\boldsymbol{h}(\boldsymbol{y}(t), t)). \tag{14}$$

where $\boldsymbol{\Phi}(\cdot) : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^{m+n}$ represents a vector array of monotonically increasing odd activation functions, and $\gamma \geq 0$ is a constant that scales the convergence rate of ZNN. The total derivative of $\boldsymbol{h}(\boldsymbol{y}(t), t)$ can be expressed as

$$\dot{\boldsymbol{h}}(\boldsymbol{y}(t), t) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}\dot{\boldsymbol{y}} + \frac{\partial \boldsymbol{h}}{\partial t}, \tag{15}$$

where

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}} = \begin{bmatrix} \frac{\partial}{\partial \dot{\boldsymbol{\theta}}}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}}\right)^\top & \frac{\partial}{\partial \boldsymbol{\lambda}}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}}\right)^\top \\ \left(\frac{\partial}{\partial \dot{\boldsymbol{\theta}}}\left(\frac{\partial L}{\partial \boldsymbol{\lambda}}\right)^\top\right)^\top & 0 \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)},$$

$$\frac{\partial \boldsymbol{h}}{\partial t} = \begin{bmatrix} \frac{\partial h_1}{\partial t} & \frac{\partial h_2}{\partial t} & \cdots & \frac{\partial h_{m+n}}{\partial t} \end{bmatrix}^\top \in \mathbb{R}^{(m+n)\times 1}.$$

The ZNN model (14) can be reformulated using (15) as

$$\dot{\boldsymbol{y}} = -\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}\right)^\dagger \left(\gamma \boldsymbol{\Phi}(\boldsymbol{h}(\boldsymbol{y}(t), t)) + \frac{\partial \boldsymbol{h}}{\partial t}\right). \tag{16}$$

where $\dagger$ represents the pseudoinverse. The neuronal form of the proposed model (16) can be written as

$$y_i = \int \sum_{j=1}^{r} \tilde{H}_{ij}\left(\gamma \phi\left(h_j\right) + \frac{\partial h_j}{\partial t}\right), \tag{17}$$

where $r = m + n$. h_i represents the i^{th} neuron. $\tilde{H}_{ij}$ represents ij^{th} component of the matrix $-\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}\right)^\dagger$ which acts as a weight for the neural network. The neural network architecture of the proposed model (16) is presented in Fig. 1. From (17) and Fig. 1, it can be seen that the neural architecture representing the proposed model (16) involves $n^2 + 2mn + m^2 + m + n - 1$ additions/subtractions, $n^2 + 2mn + m^2 + 2m + n$ multiplications, $4m + 5n$ nonlinear operations and $m + n$ integral operations. In contrast, the ZNN model proposed in [14] requires $18n^2 + 9mn + m - n$ additions/subtractions, $18n^2 + 9mn + m + 2n$ multiplications, $m + 2n$ nonlinear operations, and $3n$ integrator operations. While as the VP-ZNN model proposed in [24] requires $3(m^2 + n^2 + m_a^2) + 6(mn + m_a n + m_a m)$ additions/subtractions, $3(m^2 + n^2 + m_a^2) + 6(mn + m_a n + m_a m) + m + n + m_a$ multiplications, $m + n + m_a$ nonlinear operations, and $m + n + m_a$ integrator operations, where m_a is the number of faulty joints. Despite

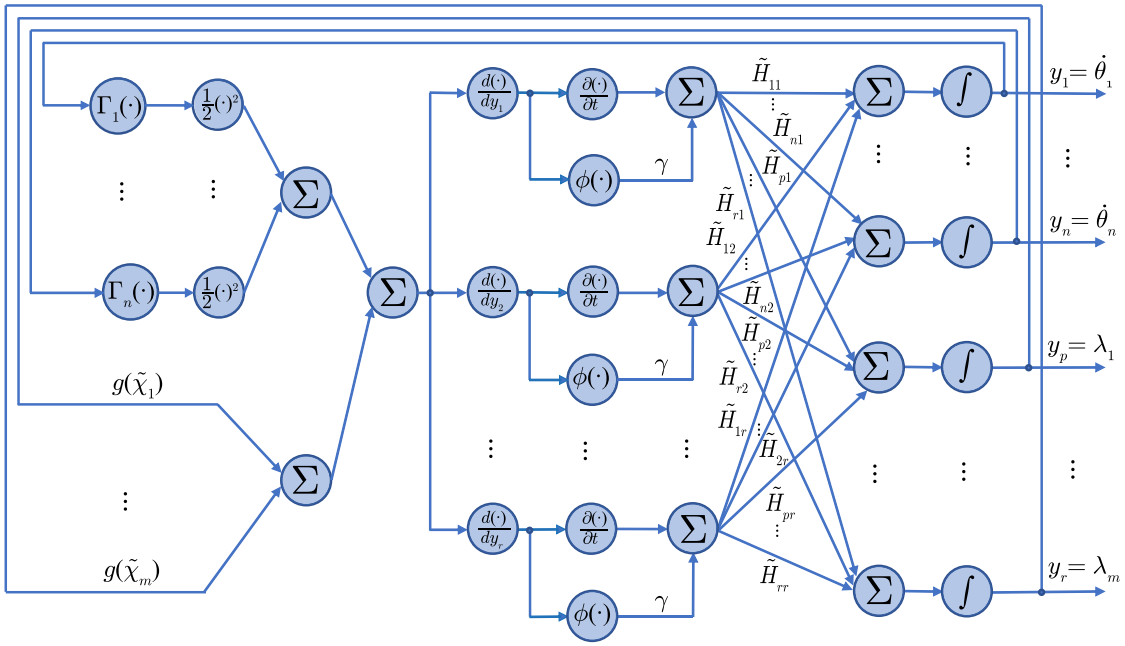


FIGURE 1. Neural network architecture for the proposed model (16).

the higher number of nonlinear transformations, the number of addition/subtraction, multiplication and integration operations is notably fewer than the ZNN architecture developed in [14] and the fault-tolerant scheme proposed in [24]. Thus, the proposed model (16) is more computationally efficient compared to the studies in [14] and [24].

C. THEORETICAL ANALYSIS

In this subsection, the theoretical analysis of the proposed model (16) is undertaken to prove the global convergence property of the system (10). To this end, we now have the following Lemmas.

Lemma 1: The optimization function $f(\boldsymbol{\chi}) = \frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\chi}$ of the problem (10) is convex with respect to the state variables $\dot{\boldsymbol{\theta}}(t)$.

Proof: The function $f(\boldsymbol{\chi})$ can be expanded as

$$f = \frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\chi} = \frac{1}{2}\sum_{k=1}^{n}\chi_k^2, \tag{18}$$

so that for ij^{th} element of the Hessian $\frac{\partial^2 f}{\partial\dot{\boldsymbol{\theta}}\partial\dot{\boldsymbol{\theta}}^\top}$, we have,

$$\frac{\partial^2 f}{\partial\dot{\theta}_j\partial\dot{\theta}_i} = \frac{1+(2\eta_i-1)ln\left(\frac{\eta_i}{1-\eta_i}\right)}{(\dot{\theta}_i^+-\dot{\theta}_i^-)^2(\eta_i(1-\eta_i))^2}\delta_{ij}, \tag{19}$$

where δ_{ij} represents the Kronecker delta function. Note that $\eta_i \in (0,1)\ \forall\ \chi_i \in (-\infty,\infty)$, so that

$$\eta_i \in \left(0,\frac{1}{2}\right) \implies (2\eta_i-1) < 0,\ ln\left(\frac{\eta_i}{1-\eta_i}\right) < 0$$
$$\eta_i \in \left(\frac{1}{2},1\right) \implies (2\eta_i-1) > 0,\ ln\left(\frac{\eta_i}{1-\eta_i}\right) > 0$$
$$\implies \begin{cases}(2\eta_i-1)ln\left(\frac{\eta_i}{1-\eta_i}\right)=0, & \text{if } \eta_i=\frac{1}{2}\\ (2\eta_i-1)ln\left(\frac{\eta_i}{1-\eta_i}\right)>0, & \text{otherwise.}\end{cases} \tag{20}$$

From (19) and (20), it can be concluded that $\frac{\partial^2 f}{\partial\dot{\theta}_i^2} > 0 \quad \forall\ i = 1,..,n$. Hence, the Hessian $\frac{\partial^2 f}{\partial\dot{\boldsymbol{\theta}}\partial\dot{\boldsymbol{\theta}}^\top}$ is a diagonal matrix with positive diagonal elements and hence is positive definite. Thus, the function f is convex with respect to $\dot{\boldsymbol{\theta}}(t)$ [33].

Lemma 2: The constraint function $\boldsymbol{g}(\tilde{\boldsymbol{\chi}}) = \frac{1}{2}\tilde{\boldsymbol{\chi}} \circ \tilde{\boldsymbol{\chi}}$ of the optimization problem (10) is convex with respect to the state variable $\dot{\boldsymbol{\theta}}(t)$.

Proof: The component-wise constraint equation is given as

$$g_k = \frac{1}{2}\tilde{\chi}_k^2,\ \forall\ 1 \le k \le m. \tag{21}$$

The ij^{th} element of the Hessian $\frac{\partial^2 g_k}{\partial\tilde{\boldsymbol{\eta}}\partial\tilde{\boldsymbol{\eta}}^\top} \in \mathbb{R}^{m\times m}$ is then given as

$$\frac{\partial^2 g_k}{\partial\tilde{\eta}_j\partial\tilde{\eta}_i} = \frac{1+(2\tilde{\eta}_k-1)ln\left(\frac{\tilde{\eta}_k}{1-\tilde{\eta}_k}\right)}{(\rho_r-\rho_l)^2(\tilde{\eta}_k(1-\tilde{\eta}_k))^2}\delta_{ijk}, \tag{22}$$

where δ_{ijk} represents the Kronecker delta function. Since $\tilde{\chi}_k$ is uniformly bounded and thus finite with $\tilde{\chi}_k \in (-\infty,\infty)$, hence $\tilde{\eta}_k \in (0,1)$. Proceeding in the same manner as Lemma 1, it can be shown that $\frac{\partial^2 g_k}{\partial\tilde{\eta}_k^2} > 0 \quad \forall\ 1 \le k \le m$. Thus, the Hessian $\frac{\partial^2 g_i}{\partial\dot{\boldsymbol{\theta}}\partial\dot{\boldsymbol{\theta}}^\top} \in \mathbb{R}^{n\times n}$ is given as

$$\frac{\partial^2 g_k}{\partial\dot{\boldsymbol{\theta}}\partial\dot{\boldsymbol{\theta}}^\top} = \boldsymbol{J}^\top\frac{\partial^2 g_k}{\partial\tilde{\boldsymbol{\eta}}\partial\tilde{\boldsymbol{\eta}}^\top}\boldsymbol{J}, \tag{23}$$

which represents a positive semi-definite matrix. Hence, the constraint functions $\boldsymbol{g}(\tilde{\boldsymbol{\chi}})$ is convex with respect to $\dot{\boldsymbol{\theta}}(t)$ [33].

Theorem 1: Given that the solution $\dot{\boldsymbol{\theta}}^*(t)$ exists within the given bounded limits, the state $\dot{\boldsymbol{\theta}}(t)$ of the system will converge to the solution $\dot{\boldsymbol{\theta}}^*(t)$ for the proposed model (16).

Proof: Lemmas 1 and 2 prove that both the objective function as well as the equality constraints for the problem (10) are convex with respect to $\dot{\boldsymbol{\theta}}(t)$. Hence, (10) represents a convex optimization problem in $\dot{\boldsymbol{\theta}}(t)$.

Consider the component-wise Lyapunov function, which is defined as

$$V(h_i) = \frac{1}{2}h_i^2(t). \tag{24}$$

The time-derivative of the Lyapunov function is given as

$$\dot{V} = h_i(t)\dot{h}_i(t) = -\gamma h_i(t)\phi(h_i) \begin{cases} < 0, & \text{if } h_i \neq 0 \\ = 0, & \text{if } h_i = 0. \end{cases} \tag{25}$$

This proves the negative definiteness of $\dot{V}$. Hence, as per the Lyapunov theory [34], all the elements of $\boldsymbol{h}(\boldsymbol{y}(t), t)$ are globally convergent to zero. Therefore, the proposed model solves the problem (10). Hence, the constraint equations have to be satisfied as

$$\lim_{t\to\infty} g_i = \lim_{t\to\infty} \tilde{\chi}_i = \lim_{t\to\infty} e_i(t) = 0 \;\forall\, 1 \leq i \leq m, \tag{26}$$

where the last equality is satisfied by invoking the fact that $\rho_l(t) = -\rho_r(t)$. From (6), as $k \geq 0$, it follows that $\lim_{t\to\infty} \boldsymbol{\epsilon}(t) = \lim_{t\to\infty} \dot{\boldsymbol{\epsilon}}(t) = \mathbf{0}$.

Let $\boldsymbol{\chi}^*$ be the value of the unconstrained variable which satisfies the given system (2) such that

$$\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}^*)(t) - \dot{\boldsymbol{r}}_d = \mathbf{0}. \tag{27}$$

Hence, it follows that

$$\lim_{t\to\infty} \boldsymbol{J}(\boldsymbol{\theta}) \left(\boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}) - \boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}^*)\right) = \mathbf{0}. \tag{28}$$

Thus, we have $\lim_{t\to\infty} \boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}) = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi}^*)$, or $\lim_{t\to\infty} \boldsymbol{\chi}(t) = \boldsymbol{\chi}^*(t)$ where the last equality is derived by invoking the one-to-one correspondence of the map $\boldsymbol{\Gamma}(\dot{\boldsymbol{\theta}})$. Note that the mapping $\boldsymbol{\Gamma}(\dot{\boldsymbol{\theta}})$ is invertible so that the inverse mapping $\boldsymbol{\Gamma}^{-1}(\boldsymbol{\chi})$ is well defined for all time. Moreover, it is apparent that the unconstrained variable $\boldsymbol{\chi}(t)$ resulting from the proposed model (16) remains uniformly bounded as $\dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}}(t) \leq \dot{\boldsymbol{\theta}}^+ \;\forall\, t \geq 0$. Hence, corresponding to the unconstrained variable $\boldsymbol{\chi}^*(t)$, there exists a unique state $\dot{\boldsymbol{\theta}}^*(t)$ satisfying $\lim_{t\to\infty} \dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{\theta}}^*(t)$, so that $\lim_{t\to\infty} \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{r}}_d$, and $\dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}}(t) \leq \dot{\boldsymbol{\theta}}^+ \;\forall\, t \geq 0$.

Furthermore, as the variable $\tilde{\chi}_i$ is uniformly bounded, we have $\rho_l(t) \leq e_i(t) \leq \rho_r(t) \;\forall 1 \leq i \leq m,\ t \in [0, \infty)$. This proves that the proposed model (16) is able to find the solution of the underdetermined system (2) subject to bounds (3), which satisfies the prescribed performance constraints.

IV. VALIDATION

In this section, numerical and experimental validation studies are undertaken with the proposed model (16) to accomplish trajectory tracking on the KUKA LBR IIWA 14 R820 manipulator with physical constraints on the joint velocity. A detailed performance comparison study with leading alternative designs is also undertaken.

A. NUMERICAL RESULTS

In this subsection, path tracking simulations are performed for the cardioid and tricuspid paths using the proposed model (16). The simulations are undertaken for both the fault-free and fault-tolerant scenarios. Then, the effect of the inclusion of prescribed performance constrains (PPCs) is demonstrated, followed by a performance comparison of the proposed model (16) with the ZNN model proposed in [14] and the VP-ZNN model proposed in [24]. The simulation results are obtained using MATLAB® 2022a.

1) FAULT-FREE CASE

In this subsection, path tracking simulations are performed for the cardioid and tricuspid paths using the proposed model (16) for the fault-free scenario. The simulations are carried out for a duration of 10 s. The limits on the joint velocity are specified as

$$\dot{\boldsymbol{\theta}}^+ = -\dot{\boldsymbol{\theta}}^- = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix}^\top \text{ rad/s}$$

The initial joint positions are specified as $\boldsymbol{\theta}(0) = [-0.26, 0.29, 0.59, -0.99, 0.49, 0.31, 0.86]$ rad. The simulations are performed with a linear activation function, $\gamma = 100$, $\delta = 100$, $\rho_0 = 2$, $k = 10$ and $\rho_\infty = 0.005$. Figs. 2 and 3 show the simulation results for the cardioid and the tricuspid path, respectively. In particular, Figs. 2a and 3b demonstrate the efficacy of the proposed model (16) in the trajectory tracking of the desired shapes while ensuring that the states of the system remain within the specified bounds (Figs. 2c and 3c). The tracking errors are of the order 10^{-5} m for both cases, as demonstrated by Figs. 2b and 3b.

To further demonstrate the capabilities of the proposed scheme (16), nonzero initial tracking errors are considered. In this scenario, it is assumed that the desired initial position is different from the actual initial position of the end-effector. The initial tracking errors are considered to be $[-4, 3.5, 2.5]\times 10^{-2}$ m and $[3.3, -3.5, -2.6]\times 10^{-2}$ m for the tricuspid and cardioid path respectively which accounts for a significant initial deviation of $[19.8\%, 17.3\%, 21.4\%]$ and $[16.3\%, 17.3\%, 22.2\%]$ compared to the size of the aforementioned shapes in the Cartesian space. The gain for position level error is specified as $k = 1$ and $k = 2$ for the cardioid and tricuspid shapes respectively. The joint velocity limits for this scenario are taken as

$$\dot{\boldsymbol{\theta}}^+ = -\dot{\boldsymbol{\theta}}^- = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \end{bmatrix}^\top \text{ rad/s}$$

The rest of the parameters are kept the same. The simulation results are shown in Figs. 4 and 5. Figs. 4a and 5a

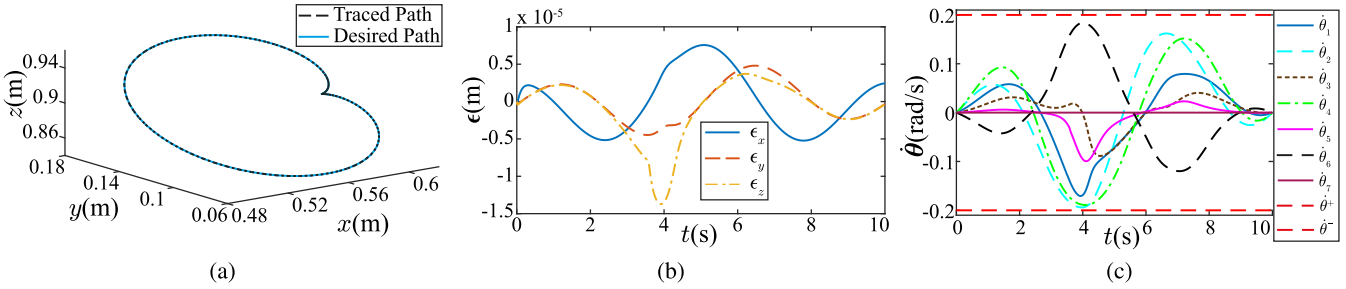


FIGURE 2. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 10$, $\delta = 100$, and $\gamma = 100$ for the fault-free case with zero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

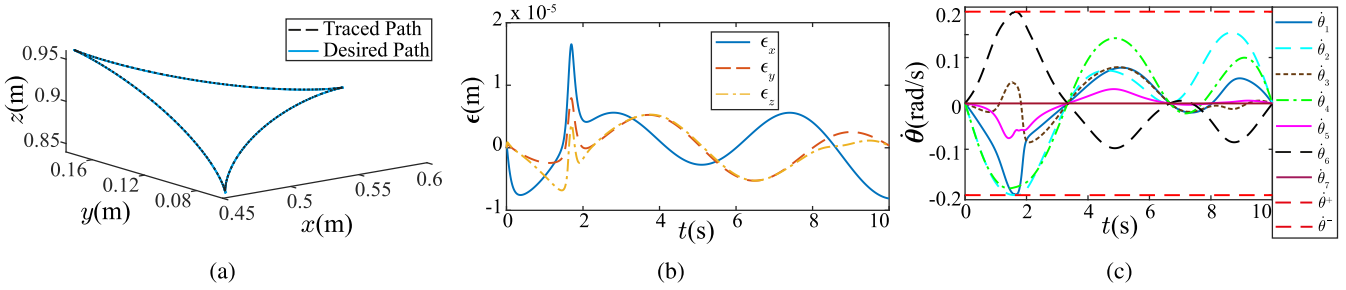


FIGURE 3. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 10$, $\delta = 100$, and $\gamma = 100$ for the fault-free case with zero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

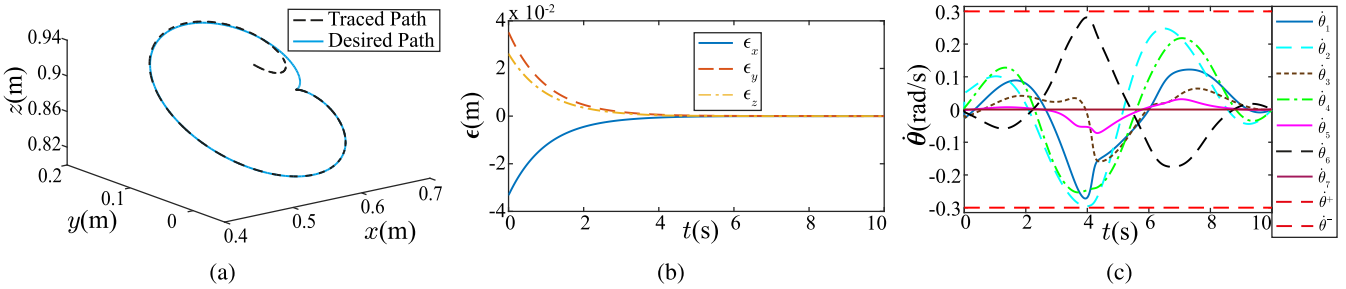


FIGURE 4. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 1$, $\delta = 100$, and $\gamma = 100$ for the fault-free case and nonzero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

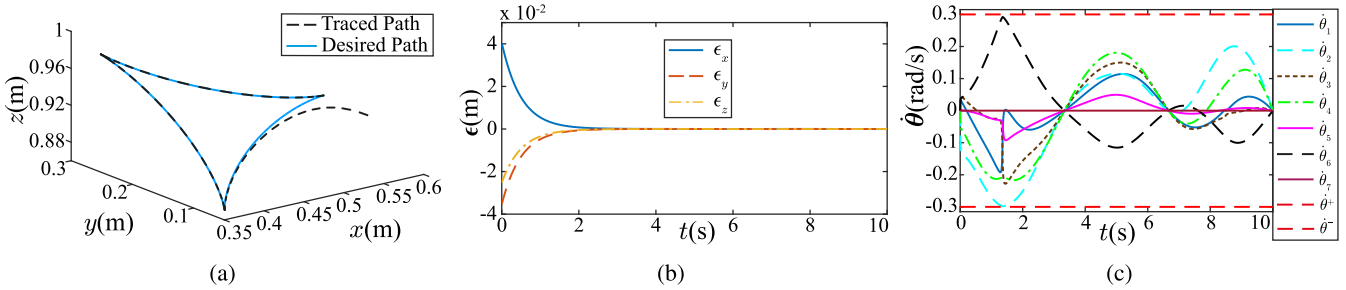


FIGURE 5. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 2$, $\delta = 100$, and $\gamma = 100$ for the fault-free case and nonzero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

demonstrate the efficacy of the proposed model (16) in tracking the desired shapes while ensuring that the states of the system remain within the specified bounds (Figs. 4c and 5c). The evolution of tracking errors is shown by Figs. 4b and 5b. These tracking errors eventually converge to an order of 10^{-5} m. As is evident from the simulations, an additional

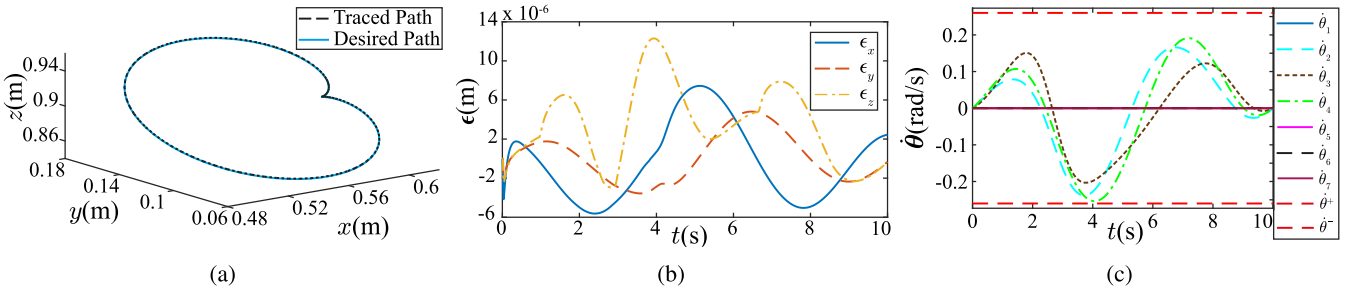


FIGURE 6. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 10$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and zero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

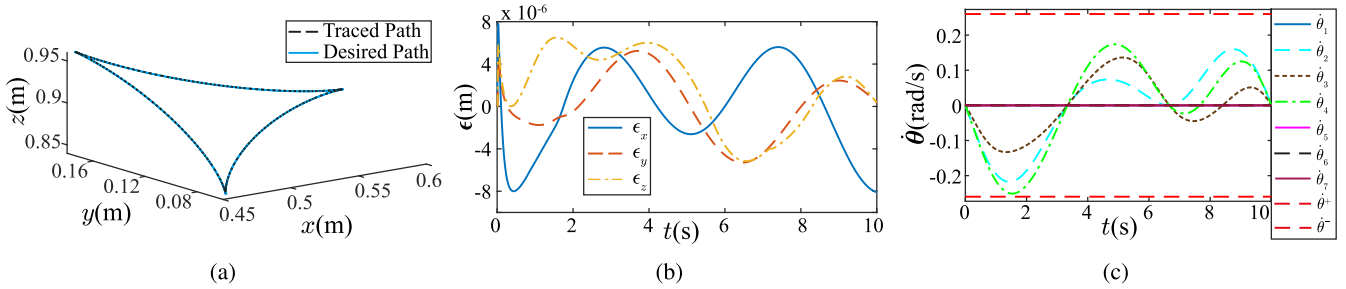


FIGURE 7. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 10$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and zero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

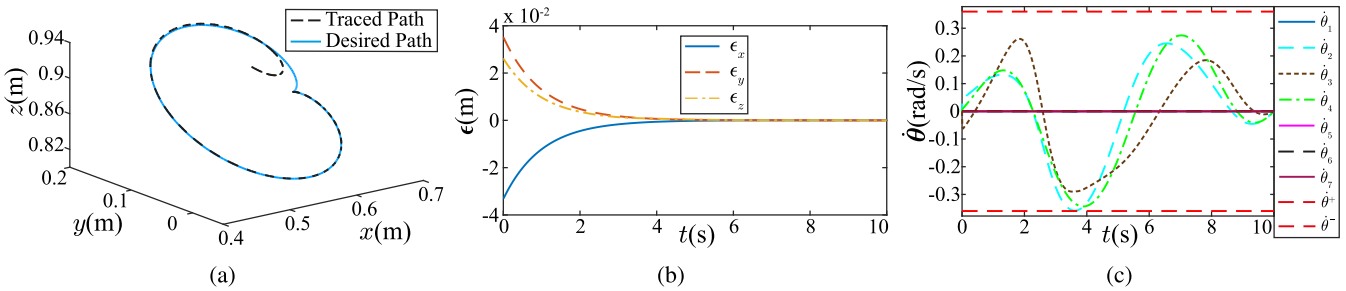


FIGURE 8. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 1$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and nonzero initial tracking errors.(a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

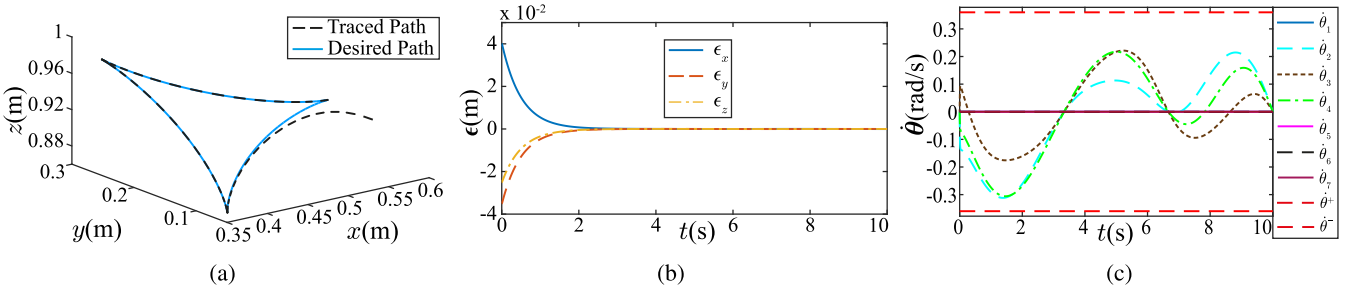


FIGURE 9. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 2$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and nonzero initial tracking errors.(a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

advantage of the proposed model (16) is that it ensures the return of the joint variables to zero values at end of the simulation run, thus rendering the proposed approach drift-free. This property of the proposed model (16) can be attributed to incorporating nonlinear mapping for imposing state constraints within the optimization framework, which gives preference to solutions that lie in the middle of the admissible region.

2) FAULT-TOLERANT CASE

In this subsection, path tracking simulations are performed using the KUKA LBR IIWA 14 R820 manipulator for the cardioid and tricuspid paths using the proposed model (16) in the presence of multiple joint failures.

To illustrate the fault tolerance capabilities of the proposed model (16), the 1^{st}, 5^{th}, and 6^{th} joints are considered to be faulty. To allow for fault tolerance, the joint velocity limits for the scenario with zero initial tracking errors are set as

$$\dot{\boldsymbol{\theta}}^{+} = -\dot{\boldsymbol{\theta}}^{-} = [10^{-4}\ 0.26\ 0.26\ 0.26\ 10^{-4}\ 10^{-4}\ 0.26]^{\top}\ \text{rad/s}$$

For the faulty joints (joints 1, 5, and 6), note that the joint velocity limits are set to very small values as $[-10^{-4},\ 10^{-4}]$ rad/s such that the motion for these joints is practically nonexistent. Figs. 6 and 7 show the tracking simulation results for fault tolerance with the faulty joints. In particular, Figs. 6a and 7a demonstrate the efficacy of the proposed model (16) in the trajectory tracking of the desired shapes in spite of the three joints being locked. Further, it is ensured that the joint velocities remain within the specified bounds (Figs. 6c and 7c). The tracking errors are of the order 10^{-5} m and 10^{-6}m, respectively (Figs. 6b and 7b) which is comparable to those obtained for the fault-free scenario.

For the scenario with nonzero initial errors, the joint velocity limits are set as

$$\dot{\boldsymbol{\theta}}^{+} = -\dot{\boldsymbol{\theta}}^{-} = [10^{-4}\ 0.36\ 0.36\ 0.36\ 10^{-4}\ 10^{-4}\ 0.36]^{\top}\ \text{rad/s}$$

The simulation results are shown in Figs. 8 and 9. Figs. 8a and 9a demonstrate the efficacy of the proposed model (16) in accomplishing trajectory tracking of the desired shapes in presence of multiple joint failures. Figs. 8c and 9c show that the joint velocities remain well within the specified limits. The tracking errors evolution for both cases is shown by Figs. 8b and 9b. These tracking errors eventually converge to an order of 10^{-5} m, which is again comparable to the fault-free case.

3) EFFECT OF PERFORMANCE CONSTRAINTS

This subsection focuses on the effect of the inclusion of prescribed performance constraints (PPCs) in the proposed scheme (16). A comparison of the average tracking error for the proposed model (16) with and without the PPCs is made using the normalized metric defined over the duration of simulation as

$$\epsilon_m = \sqrt{\frac{1}{30}\int_0^{10} ||\boldsymbol{\epsilon}(t)||_2^2\, dt}. \tag{29}$$

where $||.||_2$ represents the Euclidean norm. The comparison of the mean tracking error (ϵ_m) for the proposed scheme (16) in presence and absence of PPCs is shown in Table 1. It is observed that the mean tracking errors in the absence of PPCs exceed the corresponding values in the presence of PPCs by an order of 10^3 m for the fault-free case and 10^4 m for the fault-tolerant scenario for the case of zero initial tracking errors. For the case with nonzero initial tracking errors, mean tracking errors in the absence of PPCs still significantly exceed the corresponding values in the presence of PPCs for the case with multiple joint failures. However, the apparent difference is not as pronounced as before due to the presence of nonzero initial tracking errors which contribute significantly to the mean tracking error values. To glean a deeper insight into the effect of the PPCs for the case with nonzero initial tracking errors, the corresponding simulation results for trajectory tracking in the absence of PPCs are shown in Figs. 10 - 13. It can be seen (Figs. 10b - 13b) that the absence of prescribed performance constraints consistently results in a significant overshoot and a non-exponential decay, especially for the fault-tolerant case, thus leading to significant degradation of the transient performance. Further, as can be seen from Figs. 10c - 13c, absence of PPCs leads to a slower decay rate and larger steady-state errors which results in a poorer steady-state performance. The performance difference is drastic in some cases, as seen in Fig. 12a, where the traced path shape is severely distorted. The superior tracking error obtained in the presence of performance constraints is due to the fact that PPCs force the residual error of the system to exponentially converge to a small residual set at a decay rate equal to or faster than a predefined rate, forcing the tracking error to decay faster and achieve lower steady-state values. Thus, the inclusion of prescribed performance constraints significantly improves both the transient as well as the steady-state performance of the proposed scheme (16).

TABLE 1. Comparison of mean tracking error $\epsilon_m(\times 10^{-4})$ (m) with and without the performance constraints.

Path	Proposed (16)(PPCs)		Proposed (16) (No PPCs)	
Zero initial tracking errors				
	Fault Free	Fault Tolerant	Fault Free	Fault Tolerant
Cardioid	0.0037	0.0041	2.655	28.92
Tricuspid	0.0038	0.0038	3.460	21.61
Nonzero initial tracking errors				
	Fault Free	Fault Tolerant	Fault Free	Fault Tolerant
Cardioid	71.10	71.10	73.20	104.83
Tricuspid	54.37	54.34	56.70	69.23

4) COMPARISON WITH RELATED STUDIES

This subsection includes the performance comparison studies for the proposed model (16) with the ZNN model [14] and the VP-ZNN model [24]. This is followed by a qualitative comparison study with previous related studies to show the novelty of present work.

The performance comparison studies are only undertaken for scenarios with zero initial tracking error as the ZNN model in [14] implements trajectory tracking at the velocity level which cannot compensate for nonzero initial tracking errors. Figs. 14 and 15 present the trajectory tracking results obtained with the ZNN model [14] and the VP-ZNN model [24] for the fault-free case respectively. To illustrate the benefits of the proposed schemes for robotic trajectory tracking applications, a comparison of the average tracking error (29) and the average control effort required

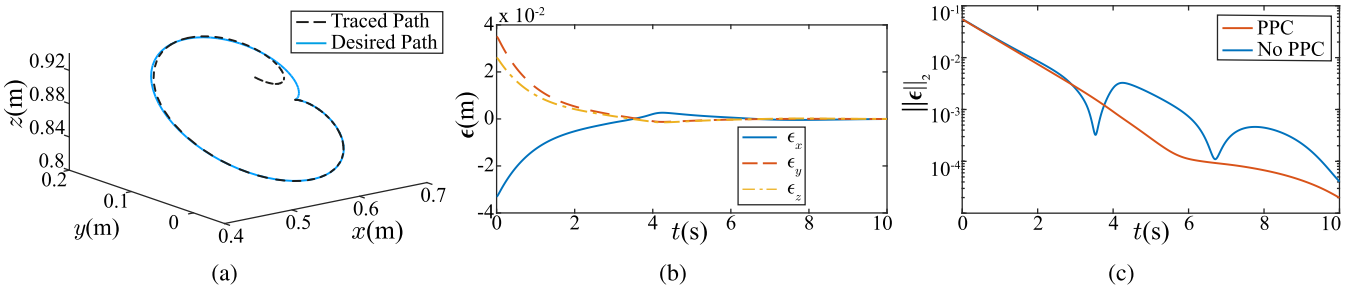


FIGURE 10. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 1$, $\delta = 100$, and $\gamma = 100$ for the fault-free case with nonzero initial tracking errors without PPCs. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Effect of PPCs on norm of tracking errors $|||\epsilon||_2$.

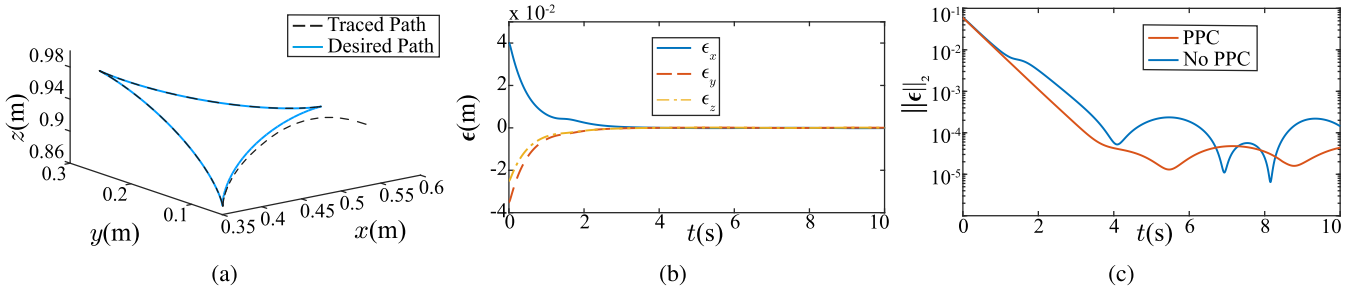


FIGURE 11. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 2$, $\delta = 100$, and $\gamma = 100$ for the fault-free case with nonzero initial tracking errors without PPCs. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Effect of PPCs on norm of tracking errors $|||\epsilon||_2$.

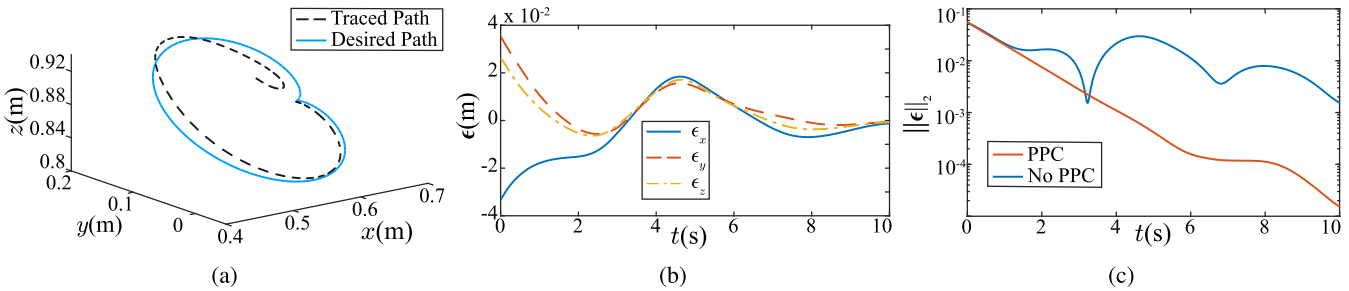


FIGURE 12. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a cardioid path for the proposed model (16) with $k = 1$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and nonzero initial tracking errors without PPCs. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Effect of PPCs on norm of tracking errors $|||\epsilon||_2$.

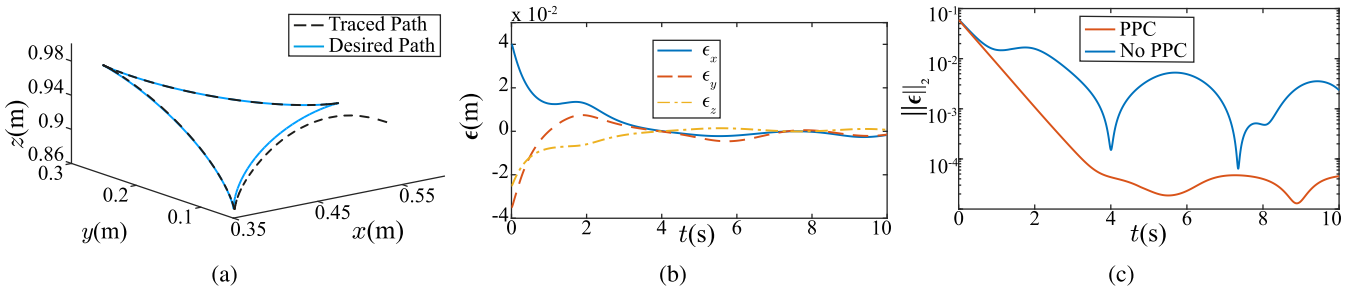


FIGURE 13. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 2$, $\delta = 100$, and $\gamma = 100$ with joint failures of 1^{st},5^{th}, and 6^{th} joint and nonzero initial tracking errors without PPCs. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Effect of PPCs on norm of tracking errors $|||\epsilon||_2$.

to accomplish trajectory tracking is considered for the ZNN [14], VP-ZNN [24] and the proposed model (16). The average control effort is defined over the duration of simulation using normalized metrics as

$$u = \sqrt{\frac{1}{70}\int_0^{10} ||\dot{\boldsymbol{\theta}}(t)||_2^2\, dt}.$$

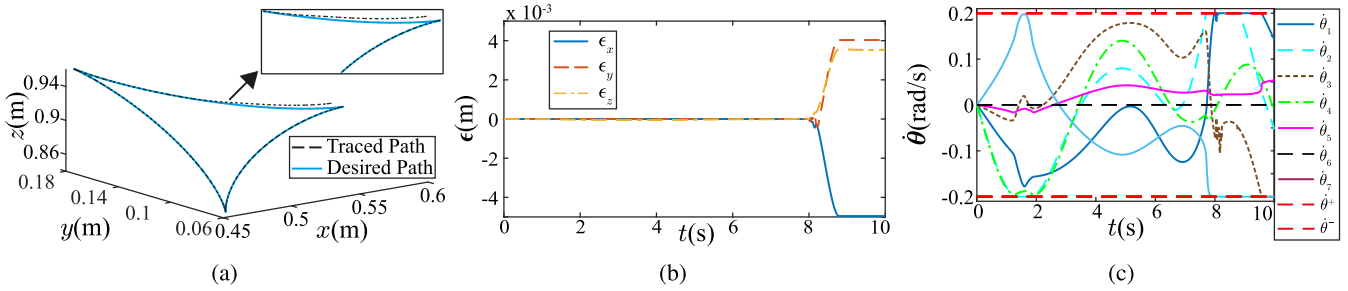


FIGURE 14. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for ZNN [14] for the fault-free case with zero initial tracking errors. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

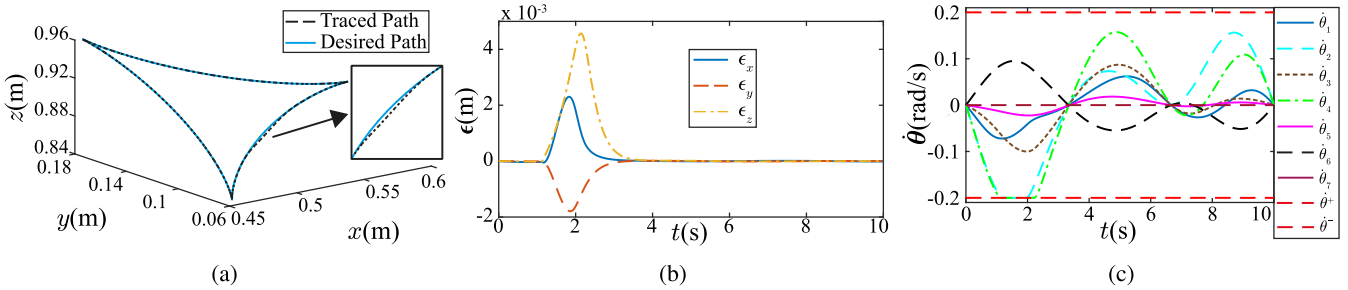


FIGURE 15. Simulation results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for VP-ZNN [24] for the fault-free case with zero initial tracking error. (a) Simulated motion path. (b) Variation of tracking error $\epsilon(t)$. (c) Variation of joint velocities $\dot{\theta}(t)$.

TABLE 2. Comparison of mean control effort u (rad/s) and mean tracking error ϵ_m (m) for various control schemes.

Path	ZNN [14]	Proposed (16)	VP-ZNN [24]
	Fault-free (u, ϵ_m ($\times 10^{-4}$))		
Cardioid	0.077, 0.117	0.071, 0.037	0.067, 10.7
Tricuspid	0.100, 16.1	0.071, 0.038	0.066, 7.35
	Fault-tolerant (u, ϵ_m ($\times 10^{-4}$))		
Cardioid	0.078, 0.325	0.078, 0.041	0.078, 0.345
Tricuspid	0.071, 0.424	0.071, 0.038	0.071, 0.322

Table 2 lists the comparison of the mean control effort and the mean tracking error obtained using the various models for the fault-free case and the fault-tolerant case with multiple joint failures and zero initial tracking errors. It is apparent that the proposed model (16) has a lower mean tracking error as compared to ZNN [14] and VP-ZNN [24], which leads to superior tracking performance. Further, it is evident that the control effort required for implementing trajectory tracking using the proposed model (16) is lower than the scheme proposed in [14] and comparable to the VP-ZNN scheme in [24]. The marginally lower computational effort required by the VP-ZNN scheme in [24] is realized at the cost of greater tracking error compared to the proposed scheme for the fault-free case, as is observed from Fig. 15. The superior performance and computational efficiency of the proposed model (16) can be attributed to the prescribed performance constraints, which drive the admissible solutions towards the middle of the admissible region, leading to a lower control effort and lower error in accomplishing trajectory tracking. This also helps avoid situations that may lead to possible input saturation scenarios when the solution ventures close to the bounded limits, as can be seen from Fig. 14. In particular, as observed from Fig. 14a, this leads to the traced path differing significantly from the desired path for the scheme proposed in [14]. This deviation from the desired path occurs due to saturation of states $\dot{\theta}_1(t)$ and $\dot{\theta}_2(t)$ (Fig. 14c) corresponding to the computed solution by the ZNN model [14]. As is apparent from Fig. 14b, this situation leads to a sudden surge in errors which leads to deviation from the desired path. A similar scenario can be seen for the VP-ZNN proposed in [24], where states $\dot{\theta}_2(t)$ and $\dot{\theta}_4(t)$ show violation of bounds as the scheme lacks any structure for enforcing velocity constraints (Fig. 15c). Again, due to saturation limits, these desired velocities (which violate the bounds constraints for this controller) cannot be achieved, leading to input saturation, as seen in Fig. 15c. Again, as before, there is a surge in tracking error, as seen in Fig. 15b, which leads to a significant deviation from the desired path. In contrast, as seen in the previous section, the proposed model (16) keeps these joint velocities away from the bounds and avoids any bound violation leading to superior steady-state performance both for the fault-free case and the case of multiple joint failures. Thus, the proposed model (16) clearly offers the advantage of rigorously combining enforcement of joint velocity constraints of ZNN [14] with the fault tolerance of VP-ZNN [24] in a computationally efficient framework.

A qualitative comparison of the proposed scheme (16) with related redundancy resolution studies is shown in Table 3. The QP formulation of the proposed scheme (16) allows it to synthesize optimal control inputs which distinguishes it from the schemes proposed in [14], [15], and [17]. The presence of

TABLE 3. Qualitative comparison of proposed scheme (16) with related studies.

Method	PPCs	State Constraints	Optimal Input	Position level Error
Proposed (16)	Yes	Yes	Yes	Yes
[14]	No	Yes	No	No
[15]	No	Yes	No	No
[16]	No	Yes	No	Yes
[17]	No	Yes	No	Yes
[23]	No	No	Yes	Yes
[35]	No	Yes	Yes	No
[36]	No	Yes	Yes	No

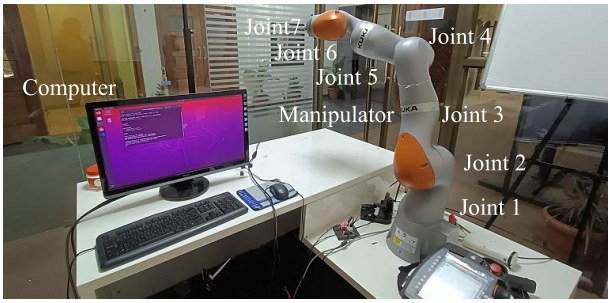


FIGURE 16. Experimental setup.

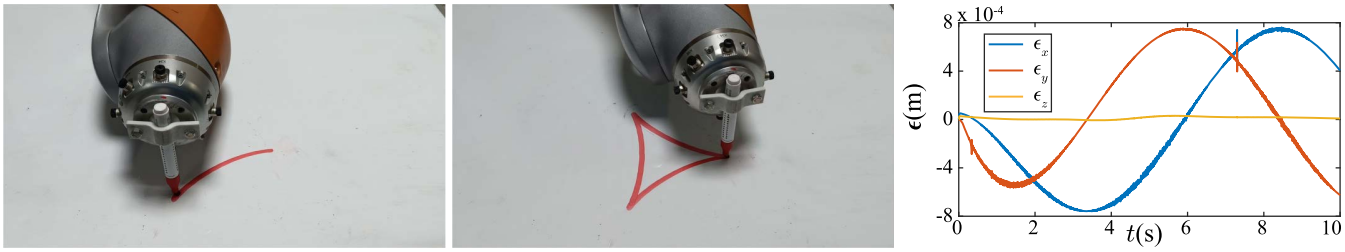
performance constraints allows the proposed scheme (16) to impose constraints over the residual error and assert control over the transient performance separating it from other optimal schemes proposed in [23], [35], and [36]. Moreover, the constraints imposed by prescribed performance constraints are dynamic in nature which distinguishes the proposed scheme (16) from [16] where the bounds imposed on the residual error are static. Further, the proposed scheme (16) employs nonlinear mapping to rigorously enforce state and input constraints, as well as prescribe performance which separates it from the previously mentioned studies. As seen from this comparison, the proposed scheme (16) is the only scheme that combines both the state and input constraints as well as prescribed performance constraints in a QP framework to synthesize an optimal input for trajectory tracking applications both in a fault-free and a fault-tolerant setting.

B. EXPERIMENTAL VERIFICATION

In this subsection, physical path tracking experiments are performed for the tricuspid and star paths using the proposed model (16) for both the fault-free and fault-tolerant scenarios with the KUKA LBR IIWA 14 R820 manipulator, as shown in Fig. 16, with zero initial tracking errors. The hardware capabilities of the setup used include the Intel® Core™ i7-7700 processor with 32GB RAM and 1GB Intel® HD Graphics 630 graphics card. The experiments are performed for a time duration of 10 s. The initial joint positions are specified as $\boldsymbol{\theta}(0) = [-0.299, 0.959, 0.562, -1.242, -0.249, -0.479, 0.477]^\top$ rad. The simulations are performed with a linear activation function, and for parameters chosen as $\gamma = 90$, $\delta = 90$, $\rho_0 = 2$, $\rho_\infty = 0.01$. The values of the gain parameter k for the tricuspid and star-shaped paths are set to 1 and 0 respectively.

The joint velocity limits for the fault-free case are specified as

$$\dot{\boldsymbol{\theta}}^{+} = -\dot{\boldsymbol{\theta}}^{-} = [0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8]^\top \text{rad/s}$$



(a) (b) (c)

FIGURE 17. Experimental results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 1$, $\delta = 90$, and $\gamma = 90$ for the fault-free case with zero initial tracking errors. (a), (b) Traced end-effector path. (c) Variation of tracking error $\epsilon(t)$.

(a) (b) (c)

FIGURE 18. Experimental results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a tricuspid path for the proposed model (16) with $k = 1$, $\delta = 90$, and $\gamma = 90$ with joint failures of 1^{st},5^{th} and 6^{th} joint with zero initial tracking errors.(a), (b) Traced end-effector path. (c) Variation of tracking error $\epsilon(t)$.

(a) (b) (c)

FIGURE 19. Experimental results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a star path for the proposed model (16) with $k = 0$, $\delta = 90$, and $\gamma = 90$ for the fault-free case with zero initial tracking errors. (a), (b) Traced end-effector path. (c) Variation of tracking error $\epsilon(t)$.

(a) (b) (c)

FIGURE 20. Experimental results for KUKA LBR IIWA 14 R820 manipulator for trajectory tracking of a star path for the proposed model (16) with $k = 0$, $\delta = 90$ and $\gamma = 90$ with joint failures of 1^{st},5^{th} and 6^{th} joint with zero initial tracking errors.(a), (b) Traced end-effector path. (c) Variation of tracking error $\epsilon(t)$.

It can be seen from Figs. 17 and 19 that the manipulator is able to trace both the tricuspid and star-shaped paths in a smooth manner for the fault-free case. As seen from Figs. 17c and 19c, the trajectory tracking errors for both paths are of the order of 10^{-4} m. For the fault-tolerant scenario, the 1^{st}, 5^{th}, and 6^{th} are again assumed to be faulty and the joint velocity limits are set to

$$\dot{\boldsymbol{\theta}}^{+} = -\dot{\boldsymbol{\theta}}^{-} = [10^{-4} \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 10^{-4} \;\; 10^{-4} \;\; 0.8]^{\top} \text{rad/s}$$

Again, the manipulator is able to trace smooth tricuspid and star paths even in presence of joint failures as seen in Figs. 18 and 20. The errors are again of the order of 10^{-4} m, as seen in Figs. 18c and 20c.

Clearly, the experimental results above demonstrate the efficacy of the proposed model (16) for accomplishing trajectory tracking for the fault-free case and the fault-tolerant case with several joint failures.

V. CONCLUSION

This study proposes and investigates a novel neural control architecture for accomplishing trajectory tracking of redundant manipulators with joint velocity constraints. The novelty of the proposed scheme lies in the synthesis of a quadratic program framework that combines a nonlinear state transformation with state and prescribed performance constraints to realize drift-free control policies that rigorously satisfy prescribed velocity and performance constraints. This framework allows the proposed scheme to provide formal guarantees for the delivery of stringent transient and steady-state performance. A detailed theoretical analysis is undertaken to demonstrate the stability and convergence properties of the proposed scheme (16). Complexity analysis reveals the higher computational efficiency of the proposed model compared to leading alternative designs. Computer simulations are used to verify the efficacy of the proposed model in solving the trajectory tracking problem for the KUKA LBR IIWA 14 R820 manipulator with bounds on joint velocities both in the absence and presence of joint failures. These validation studies clearly demonstrate the superior performance delivered using lower control effort for the proposed model (16) compared to the alternate designs. Moreover, the proposed model is shown to realize zero terminal joint velocities obtained by embedding the nonlinear mapping for imposing state constraints within the optimization framework. Finally, physical experiments are carried out to further substantiate the efficacy of the proposed scheme.

A possible direction for future work is the inclusion of obstacle avoidance capabilities in the proposed models to accomplish obstacle-aware trajectory tracking with bounds on the system state and control input. Moreover, incorporating a state observer for estimating the system's Jacobian online, which could potentially lead to a platform-agnostic observer-controller framework, would find wide suitability in constrained trajectory tracking applications. Finally, the efficacy of the proposed scheme for trajectory tracking applications in redundant robotic manipulators alludes to the suitability of these schemes to other constrained robotic applications such as visual target tracking and visual servo control.

REFERENCES

[1] A. Ghosal, *Robotics: Fundamental Concepts and Analysis*. London, U.K.: Oxford Univ. Press, 2006.
[2] Y. Zhang and Z. Zhang, *Repetitive Motion Planning and Control of Redundant Robot Manipulators*. Cham, Switzerland: Springer, 2014.
[3] Y. Zhang, D. Jiang, and J. Wang, "A recurrent neural network for solving Sylvester equation with time-varying coefficients," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1053–1063, Sep. 2002.
[4] Y. Zhang, L. Xiao, Z. Xiao, and M. Mao, *Zeroing Dynamics, Gradient Dynamics, and Newton Iterations*. Boca Raton, FL, USA: CRC Press, 2018.
[5] Y. Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1477–1490, Nov. 2005.
[6] L. Jin, S. Li, and B. Hu, "RNN models for dynamic matrix inversion: A control-theoretical perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 189–199, Jan. 2018.
[7] L. Jin, Y. Zhang, and S. Li, "Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2615–2627, Dec. 2016.
[8] D. Guo and Y. Zhang, "A new variant of the Zhang neural network for solving online time-varying linear inequalities," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 468, no. 2144, pp. 2255–2271, 2012.
[9] D. Guo and Y. Zhang, "Zhang neural network for online solution of time-varying linear matrix inequality aided with an equality conversion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 370–382, Feb. 2014.
[10] L. Xiao and Y. Zhang, "Two new types of Zhang neural networks solving systems of time-varying nonlinear inequalities," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 10, pp. 2363–2373, Oct. 2012.
[11] Y. Zhang, Y. Wang, L. Jin, B. Mu, and H. Zheng, "Different ZFs leading to various ZNN models illustrated via online solution of time-varying underdetermined systems of linear equations with robotic application," in *Proc. Int. Symp. Neural Netw.* Cham, Switzerland: Springer, 2013, pp. 481–488.
[12] Z. Zhang, L. Zheng, and T. Qiu, "A gain-adjustment neural network based time-varying underdetermined linear equation solving method," *Neurocomputing*, vol. 458, pp. 184–194, Oct. 2021.
[13] H. Lu, L. Jin, J. Zhang, Z. Sun, S. Li, and Z. Zhang, "New joint-drift-free scheme aided with projected ZNN for motion generation of redundant robot manipulators perturbed by disturbances," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 9, pp. 5639–5651, Sep. 2021.
[14] F. Xu, Z. Li, Z. Nie, H. Shao, and D. Guo, "New recurrent neural network for online solution of time-dependent underdetermined linear system with bound constraint," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2167–2176, Apr. 2019.
[15] Z. Ma and D. Guo, "Discrete-time recurrent neural network for solving bound-constrained time-varying underdetermined linear system," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 3869–3878, Jun. 2021.
[16] H. Lu, L. Jin, X. Luo, B. Liao, D. Guo, and L. Xiao, "RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5931–5942, Nov. 2019.
[17] Y. Kong, T. Hu, J. Lei, and R. Han, "A finite-time convergent neural network for solving time-varying linear equations with inequality constraints applied to redundant manipulator," *Neural Process. Lett.*, vol. 54, no. 1, pp. 125–144, Feb. 2022.
[18] W. Li, P. W. Y. Chiu, and Z. Li, "A novel neural approach to infinity-norm joint-velocity minimization of kinematically redundant robots under joint limits," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 21, 2021, doi: 10.1109/TNNLS.2021.3095122.
[19] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2090–2099, Oct. 2008.
[20] A. K. Kostarigka, Z. Doulgeri, and G. A. Rovithakis, "Prescribed performance tracking for flexible joint robots with unknown dynamics and variable elasticity," *Automatica*, vol. 49, no. 5, pp. 1137–1147, 2013.
[21] X. Bu, G. He, and D. Wei, "A new prescribed performance control approach for uncertain nonlinear dynamic systems via back-stepping," *J. Franklin Inst.*, vol. 355, no. 17, pp. 8510–8536, Nov. 2018.
[22] X. Bu and Q. Wang, "Neural network–based nonaffine control of air-breathing hypersonic vehicles with prescribed performance," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 1, Jan. 2018, Art. no. 172988141875524.
[23] J. Li, M. Mao, F. Uhlig, and Y. Zhang, "Z-type neural-dynamics for time-varying nonlinear optimization under a linear equality constraint with robot application," *J. Comput. Appl. Math.*, vol. 327, pp. 155–166, Jan. 2018.
[24] N. Zhong, X. Li, Z. Yan, and Z. Zhang, "A neural control architecture for joint-drift-free and fault-tolerant redundant robot manipulators," *IEEE Access*, vol. 6, pp. 66178–66187, 2018.
[25] K. Li and Y. Zhang, "Fault-tolerant motion planning and control of redundant manipulator," *Control Eng. Pract.*, vol. 20, no. 3, pp. 282–292, Mar. 2012.
[26] Z. Li, C. Li, S. Li, and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7469–7478, Dec. 2020.
[27] L. Jin, B. Liao, M. Liu, L. Xiao, D. Guo, and X. Yan, "Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network," *Frontiers Neurorobot.*, vol. 11, p. 50, Sep. 2017.
[28] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2016.
[29] Y. Zhang, S. Li, J. Gui, and X. Luo, "Velocity-level control with compliance to acceleration-level constraints: A novel scheme for manipulator redundancy resolution," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 921–930, Mar. 2018.
[30] J. Xiang, C. Zhong, and W. Wei, "A varied weights method for the kinematic control of redundant manipulators with multiple constraints," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 330–340, Apr. 2012.
[31] Y. Zhang, D. Guo, and S. Ma, "Different-level simultaneous minimization of joint-velocity and joint-torque for redundant robot manipulators," *J. Intell. Robot. Syst.*, vol. 72, no. 3, pp. 301–323, 2013.
[32] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[33] J. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Cham, Switzerland: Springer, 2010.
[34] A. H. Nayfeh and B. Balachandran, *Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods*. Hoboken, NJ, USA: Wiley, 2008.
[35] S. Shao, H. Li, S. Qin, G. Li, and C. Luo, "An inverse-free Zhang neural dynamic for time-varying convex optimization problems with equality and affine inequality constraints," *Neurocomputing*, vol. 412, pp. 152–166, Oct. 2020.
[36] D. Guo and Y. Zhang, "Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1203–1217, Oct. 2014.

RAJPAL SINGH received the Bachelor of Technology degree in mechanical engineering from the National Institute of Technology, Srinagar. He is currently pursuing the Ph.D. degree with the Indian Institute of Sciences, Bengaluru.

His research interests include the areas of nonlinear control, autonomous vision, and data driven modelling and control.

JISHNU KESHAVAN (Member, IEEE) received the B.Tech. degree in aerospace engineering from the Indian Institute of Technology Bombay, Mumbai, in 2004, and the M.S. and Ph.D. degrees in aerospace engineering from the University of Maryland, College Park, in 2007 and 2012, respectively.

He is currently an Assistant Professor in mechanical engineering with the Indian Institute of Science, Bengaluru. His research interests include the areas of dynamical systems theory, data-driven nonlinear dynamics and control, and autonomous vision.