

Article

Singularity-Free Prescribed-Time Distributed Resource Allocation Based on Time Space Deformation

Shuaiyu Zhou¹, Peng Yi², Shaofu Yang³ and Yiheng Wei^{1,*}¹ The School of Mathematics, Southeast University, Nanjing 211189, China² The Department of Control Science & Engineering, Tongji University, Shanghai 200092, China³ The School of Computer Science and Engineering, Southeast University, Nanjing 211189, China

* Correspondence: neudawei@seu.edu.cn

How To Cite: Zhou, S.; Yi, P.; Yang, S.; et al. Singularity-Free Prescribed-Time Distributed Resource Allocation Based on Time Space Deformation. *Applied Mathematics and Statistics* 2025, 2(1), 1. <https://doi.org/10.53941/ams.2025.100001>.

Received: 11 February 2025

Revised: 22 March 2025

Accepted: 25 March 2025

Published: 27 March 2025

Abstract: This paper proposes a novel singularity-free prescribed-time distributed resource allocation algorithm. By scaling fixed-time systems using the time space deformation method, the proposed algorithm avoids the singularity problem caused by time-varying high-gain functions. To make the algorithm applicable to second-order multi-agent systems, a singularity-free prescribed-time signal tracking controller is also proposed. Finally, the performance of the proposed algorithm is verified through a power allocation task based on actual wind farm data.

Keywords: distributed resource allocation; prescribed-time stability; singularity-free

1. Introduction

Distributed optimization is a common task requirement in multi-agent systems (MASs), aiming to achieve globally optimal decision-making through local information interactions [1]. It is widely applied in fields such as smart grid resource allocation [2], and unmanned aerial vehicle (UAV) formation control [3]. Among these applications, the distributed resource allocation (DRA) problem requires the efficient allocation of limited resources (such as power, bandwidth, and computing resources) among multiple agents while satisfying local constraints and global demands. For example, in wind farm power allocation, it is necessary to minimize the fatigue damage of wind turbines under real-time wind speed variations; in smart grids, it is essential to dynamically balance the load and power generation. Centralized methods rely on global information and a central node, making it difficult to cope with dynamic changes in network topology and communication delays. In contrast, distributed algorithms achieve efficient solutions through local cooperation and have become a research hotspot in the past decade.

Early distributed resource allocation algorithms were mostly based on gradient descent or dual decomposition, with convergence properties typically being asymptotic or exponential. For example, a continuous-time algorithm is proposed based on the Karush-Kuhn-Tucker (KKT) conditions, which used local gradient information to constrain the node states [4]. However, the convergence speed depended on the network topology and initial conditions. To solve the DRA problem over directed networks, a consensus-based gradient descent method is designed to achieve asymptotic convergence in balanced directed networks [5], but they could not guarantee an upper bound on the convergence time. These methods were limited in scenarios with high real-time requirements, such as dynamic power dispatch.

To improve the convergence speed, finite-time and fixed-time algorithms have been widely studied. A consensus-based approach that can obtain the optimal solution within finite time is proposed where the objective function is time-varying but the existence of the identical Hessian matrix of the global objective function is assumed [6]. However, the finite-time upper bound depends on the initial states. Hence, a series of fixed-time distributed algorithms are proposed to solve the DRA problem [7, 8]. Although fixed-time algorithms are superior to asymptotic methods, the upper bound of settling time is still reliant on coupled system parameters, which may not meet the demand for precise time control in highly dynamic environments (such as sudden load fluctuations).



In recent years, the prescribed-time convergence framework has been integrated into distributed resource allocation algorithms, where the settling time can be directly assigned or predetermined without relying on any other parameters [9]. Numerous prescribed-time resource allocation algorithms have been proposed [10–12], most of which are based on a time-dependent gain function to achieve prescribed-time stability. However, these algorithms often encounter the challenge of singularities caused by infinite gains. To address this issue, several singularity-free methods have been proposed for prescribed-time stabilization, which can serve as a foundation for designing singularity-free prescribed-time distributed resource allocation algorithms. One approach involves imposing saturation on the time-varying high gains, but this method only ensures that the system states remain within a bounded set by the prescribed time [13]. For achieving precise convergence in prescribed-time stabilization, a singularity-free method utilizing periodic delayed feedback was introduced [14]. Nevertheless, in distributed scenarios, such methods require the introduction of delays to facilitate communication among multi-agent networks. Recently, a novel singularity-free approach based on *time-space deformation* was proposed, which is delay-independent and guarantees exact convergence [15]. This method holds significant potential for adaptation in the development of singularity-free distributed resource allocation algorithms. The major contributions of this paper can be summarized as follows.

- (1) A distributed resource allocation algorithm with prescribed-time convergence is proposed. Compared with existing similar algorithms [12, 16, 17], the algorithm proposed in this paper avoids the singularity problem caused by time-varying high-gain functions based on the time space deformation method while ensures precise convergence within prescribed times, ensuring practical feasibility.
- (2) Unlike most existing distributed resource allocation algorithms [12, 18], the algorithm proposed in this paper is implemented based on a second-order multi-agent system. Specifically, a singularity-free prescribed-time signal tracking controller for second-order systems is proposed.

The rest of this paper is organized as follows. Section 2 introduces several fundamental concepts and formulates the distributed resource allocation problem. In Section 3, the proposed resource allocation algorithm and signal tracking controller are provided. Section 4 demonstrates the performance of the proposed algorithms through a limited power allocation problem. Section 5 provides the conclusion.

Notations: Throughout the paper, $\|\cdot\|$ denotes the 2-norm for vectors and the Frobenius norm for matrices. $\|\cdot\|_\infty$ denotes the ∞ -norm. I_N denotes an N order identity matrix, $\mathbf{1}_N$ denotes an N -dim vector with its components being 1 and $\mathbf{0}_N$ denotes an N -dim vector with its components being 0. \otimes represents the Kronecker product operator. \mathcal{N}_i denotes the set of neighbour nodes of agent i in a multi-agent system. For $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, define $\text{sgn}(\mathbf{x}) = [\text{sgn}(x_1), \text{sgn}(x_2), \dots, \text{sgn}(x_n)]^\top$ and $\text{sig}^\alpha(\mathbf{x}) = [\text{sig}^\alpha(x_1), \text{sig}^\alpha(x_2), \dots, \text{sig}^\alpha(x_n)]^\top$ where $\text{sgn}(\cdot)$ denotes the signum function and $\text{sig}(\cdot) = |\cdot|^\alpha \text{sgn}(\cdot)$. Denote $\mathbb{R}_{\geq x} = \{y \in \mathbb{R} : y \geq x\}$.

2. Preliminaries and Problem Formulation

2.1. Globally Prescribed-Time Stable

Definition 1. [9] Consider the system defined by

$$\dot{x}(t) = f(t, x(t)), \quad t \in \mathbb{R}_{\geq t_0}, \quad x(t_0) = x_0, \quad (1)$$

where $x \in \mathbb{R}^n$ is the state vector, $f : \mathbb{R}_{\geq t_0} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a locally uniformly bounded nonlinear vector function in time. The origin of system (1) is said to be globally prescribed-time stable if it is globally asymptotically stable and any solution $x(t)$ reaches the origin no later than a prescribed time.

2.2. Finite-Time Stability of Integrator Chain System

Lemma 1. [19] (Proposition 8.1) Consider a Hurwitz polynomial $s^n + \beta_n s^{n-1} + \dots + \beta_2 s + \beta_1$ with $\beta_1, \dots, \beta_n > 0$, and the system

$$\dot{x}_i = x_{i+1}, i = 1, \dots, n-1, \dot{x}_n = u, \quad (2)$$

where $x_i \in \mathbb{R}, i = 1, \dots, n, u \in \mathbb{R}$ is the control input. There exists $\varepsilon \in (0, 1)$ such that $\forall \alpha \in (1 - \varepsilon, 1)$, the origin is a globally finite-time stable equilibrium under the controller

$$u = -\beta_1 \text{sig}^{\alpha_1}(x_1) - \dots - \beta_n \text{sig}^{\alpha_n}(x_n), \quad (3)$$

where $\alpha_{i-1} = \frac{\alpha_i \alpha_{i+1}}{2\alpha_{i+1} - \alpha_i}, i = 2, \dots, n, \alpha_{n+1} = 1$ and $\alpha_n = \alpha$.

2.3. Graph Theory

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is defined by a node set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$, an undirected edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and an adjacency matrix $\mathcal{A} = [a_{ij}]_{N \times N}$ with non-negative elements. The degree of node v_i is denoted as $d(v_i)$ and is calculated as $d(v_i) = \sum_{j=1, j \neq i}^N a_{ij}$, where $i = 1, 2, \dots, N$. This degree is an element of the matrix $D = \text{diag}\{d(v_1), d(v_2), \dots, d(v_N)\}$. The Laplacian matrix L is defined as $D - A$.

Assumption 1. Graph \mathcal{G} is undirected and connected.

2.4. Problem Formulation

Consider a multi-agent network composed of N agents,

$$\begin{cases} \dot{x}_i(t) = y_i(t), & i \in \mathcal{V}, \\ \dot{y}_i(t) = u_i(t), \end{cases} \quad (4)$$

where x_i and $y_i \in \mathbb{R}^n$ are the local variable vectors of agent i and $u_i \in \mathbb{R}^n$ is the input protocol. The assumed graph modelling the communication network is \mathcal{G} . The goal is to solve the following resource allocation problem in a distributed manner,

$$\begin{aligned} \{x_i^*\}_{i=1}^N &:= \arg \min_{\{x_i\}_{i=1}^N} \sum_{i=1}^N f_i(x_i), \\ \text{s.t. } \sum_{i=1}^N x_i &= \sum_{i=1}^N b_i, \quad \forall i \in \mathcal{V}, \end{aligned} \quad (5)$$

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is the local cost function and x_i^* denotes the optimal solution solved locally by agent i , and b_i represents the amount of resources temporarily held by the i -th agent to be allocated. Let $\mathbf{x} = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top \in \mathbb{R}^{nN}$, $\mathbf{y} = [y_1^\top, y_2^\top, \dots, y_N^\top]^\top \in \mathbb{R}^{nN}$, $\nabla f(\mathbf{x}) = [\nabla f_1(x_1)^\top, \dots, \nabla f_N(x_N)^\top]^\top$, $\tilde{f}(\mathbf{x}) = \sum_{i=1}^N f_i(x_i)$, $x \in \mathbb{R}^n$, $\mathcal{N}_i = \{x_k \in \mathbb{R}^n : k \in \mathcal{N}_i\}$, and $\mathbf{x}^* = [x_1^{*\top}, x_2^{*\top}, \dots, x_N^{*\top}]^\top \in \mathbb{R}^{nN}$ being the optimal solution.

Assumption 2. Each local cost function $f_i(x_i)$ is twice continuously differentiable, γ_i -strongly convex with $\gamma_i > 0$.

3. Main Results

3.1. Singularity-Free Prescribed-Time Optimal Signal Generator

In this section, a distributed prescribed-time optimal signal generator is designed based on a virtual first-order multi-agent system, which is described as

$$\dot{r}_i = v_i, \quad i \in \mathcal{V}, \quad (6)$$

where $r_i \in \mathbb{R}^n$ is the local decision variable vector of agent i and $v_i \in \mathbb{R}^n$ is the input signal. The objective is to ensure that the outputs of all virtual agents converge to the optimal solution, i.e., $r_i^* = x_i^*$. Denote $\mathbf{r} = [r_1^\top, r_2^\top, \dots, r_N^\top]^\top \in \mathbb{R}^{nN}$, $\mathbf{r}^* = [r_1^{*\top}, r_2^{*\top}, \dots, r_N^{*\top}]^\top \in \mathbb{R}^{nN}$, $\mu_k(t) = \frac{T_k}{T_k^2 - (t - t_{k-1})^2}$, $\varrho_k(t) := \varrho_k(t; t_{k-1}, T_k) =$

$$\begin{cases} \mu_k(t), & t \in [t_{k-1}, t_{k-1} + T_k) \\ 0, & \text{otherwise} \end{cases}, \quad k = 1, 2, \text{ and } t_1 = t_0 + T_1. \text{ To address problem (5) without the singularity issue,}$$

a two-stage singularity-free distributed input protocol for the virtual system (6) is proposed

$$v_i(t) = v_i^s(t) + v_i^f(t), \quad (7a)$$

$$v_i^s(t) = \begin{cases} -\varrho_1(t) \mathcal{S}_{p_1, q_1}(s_i(t)), & \text{if } s_i(t) \neq \mathbf{0}_n, \\ -\mathcal{S}_{p_1, q_1}(s_i(t)), & \text{otherwise,} \end{cases} \quad (7b)$$

$$v_i^f(t) = \begin{cases} -\varrho_2(t) \mathcal{F}_{p_2, q_2}(r_i(t), \mathbf{r}_{\mathcal{N}_i}(t)), & \text{if } \tilde{c}_i(t) \neq 1, \\ -\mathcal{F}_{p_2, q_2}(r_i(t), \mathbf{r}_{\mathcal{N}_i}(t)), & \text{otherwise,} \end{cases} \quad (7c)$$

$$\dot{\phi}_i(t) = -v_i^f(t), \quad (7d)$$

$$s_i(t) = r_i(t) - b_i + \phi_i(t), \quad (7e)$$

where $k_1, k_2 > 0$, $c_1, c_2 > 0$, $0 < p_1, p_2 < 1$, $q_1, q_2 > 1$, $\mathcal{S}_{p_1, q_1}(s_i(t)) = k_1 \text{sig}^{p_1}(s_i(t)) + k_2 \text{sig}^{q_1}(s_i(t))$, $\mathcal{F}_{p_2, q_2}(r_i(t), \mathbf{r}_{\mathcal{N}_i}(t)) = c_1 \sum_{j=1}^N a_{ij} \text{sig}^{p_2}(\nabla f_i(r_i(t)) - \nabla f_j(r_j(t))) + c_2 \sum_{j=1}^N a_{ij} \text{sig}^{q_2}(\nabla f_i(r_i(t)) - \nabla f_j(r_j(t)))$, and

$$\tilde{c}_i(t) = \begin{cases} 1, & \text{if } \nabla f_i(r_i(t)) = \nabla f_j(r_j(t)), \forall j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases}$$

When $\tilde{c}_i(t) = 1$, it indicates that consensus among all local gradients of cost functions has been achieved in the multi-agent network.

Remark 1. The core idea of the proposed algorithm lies in “converge in advance before the prescribed T_i and then switch system,” where the most critical part, “converges in advance before the prescribed T_i ,” is achieved by applying the time space deformation method to a fixed-time stable system. This algorithm ensures convergence at each stage by a certain time T_i^\dagger , where $T_i^\dagger < T_i$, after which the system needs to be switched to one without time-varying high-gain functions before T_i , thereby avoiding singularity issues.

Remark 2. The designed input protocol (7) consists of a two-stage design, as shown in Figure 1.

- First stage** ($[t_0, t_0 + T_1]$): The objective is to drive the designed sliding mode variable $s_i(t)$ to converge to $\mathbf{0}_n$. This eliminates the initial value restriction commonly found in most distributed resource allocation algorithms, i.e., $\sum_{i=1}^N r_i(t_0) = \sum_{i=1}^N b_i(t_0)$.
- Second stage** ($[t_1, t_1 + T_2]$): On the sliding surface $s_i(t) = \mathbf{0}_n$, the goal is to ensure that the gradients of the agents reach consensus, enabling the virtual system states r_i to satisfy the optimality conditions of the resource allocation problem (5). This guarantees that the virtual system can find the global optimal solution within the prescribed time.

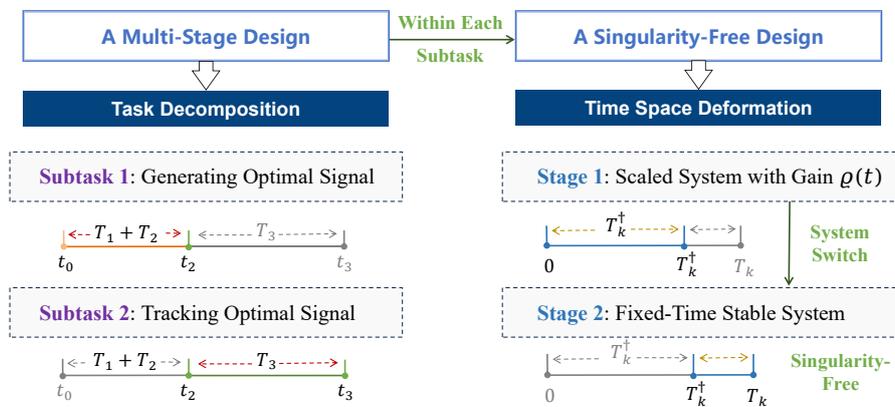


Figure 1. Multi-Stage and Singularity-Free Design.

Remark 3 (Gradient Consensus and Optimality). In (7), it can be observed that the consensus on the gradients of the loss functions among agents is required. Such a requirement is fundamentally rooted in the KKT optimality conditions of problem (5). The Lagrangian of this problem is given by $\mathcal{L}(\{x_i\}_{i=1}^N, \lambda) = \sum_{i=1}^N f_i(x_i) + \lambda(\sum_{i=1}^N b_i - \sum_{i=1}^N x_i)$, where λ is the Lagrange multiplier associated with the global constraint. The Karush-Kuhn-Tucker (KKT) optimality conditions for this problem are $\frac{\partial \mathcal{L}}{\partial x_i} = \nabla f_i(x_i^*) - \lambda^* = 0 \Rightarrow \nabla f_i(x_i^*) = \lambda^* (\forall i \in \mathcal{V})$. The stationarity condition $\nabla f_i(x_i^*) = \lambda^* (\forall i \in \mathcal{V})$ implies that all agents must share identical marginal costs at optimality. This gradient consensus ensures Pareto efficiency, where no agent can reduce its cost without increasing others’ costs.

Theorem 1. Under Assumptions 1 and 2, system (6) with protocol (7) is globally prescribed-time stable from any initial states. The trajectories of all agents in the virtual multi-agent system converge to the optimal solution of problem (5) within the prescribed-time interval $T_1 + T_2$, i.e., $\lim_{t \rightarrow (t_0 + T_1 + T_2)^-} r_i(t) = x_i^*$ and $r_i(t) \equiv x_i^*, \forall t \in [t_0 + T_1 + T_2, +\infty)$.

Proof. The following proof mainly employs the time space deformation approach.

Stage I: Letting

$$\tau_1(t; T_1) = \frac{1}{2} \ln \frac{T_1 + t - t_0}{T_1 - (t - t_0)} : [t_0, t_0 + T_1) \rightarrow \mathbb{R}_{\geq 0}, \tag{8}$$

one has $\dot{\tau}_1(t; T_1) = \mu_1(t)$ and $\mu_1(t) = \frac{(e^{\tau_1} + e^{-\tau_1})^2}{4T_1}$. Moreover, since (8) is invertible, its inverse can be shown as follows,

$$t(\tau_1; T_1) = t_0 + T_1 \tanh(\tau_1) : \mathbb{R}_{\geq 0} \rightarrow [t_0, t_0 + T_1). \tag{9}$$

To implement the time space deformation in (7), the following state transformations are applied,

$$\bar{s}_i(\tau_1) = s_i(t(\tau_1; T_1)). \tag{10}$$

By differentiating (10) along the solutions of (6) and (7) when $t_0 \leq t < t_0 + T_1$ and $s_i(t) \neq \mathbf{0}_n$, the following scaled model is derived using (9),

$$\begin{aligned} \dot{\bar{s}}_i(\tau_1) &= \frac{ds_i(t(\tau_1;T_1))}{dt(\tau_1;T_1)} \frac{\partial t(\tau_1;T_1)}{\partial \tau_1} \\ &= -\mathcal{S}_{p_1,q_1}(\bar{s}_i(\tau_1)) \\ &= -k_1 \text{sig}^{p_1}(\bar{s}_i(\tau_1)) - k_2 \text{sig}^{q_1}(\bar{s}_i(\tau_1)). \end{aligned} \tag{11}$$

Define the Lyapunov function candidate $V_i(\tau_1) = \frac{1}{2} \bar{s}_i^\top(\tau_1) \bar{s}_i(\tau_1)$. Differentiating V_i along (11) yields

$$\begin{aligned} \dot{V}_i(\tau_1) &= -k_1 \text{sig}^{p_1+1}(\bar{s}_i(\tau_1)) - k_2 \text{sig}^{q_1+1}(\bar{s}_i(\tau_1)) \\ &= -k_1 2^{\frac{1+p_1}{2}} V_i^{\frac{1+\alpha}{2}}(\tau_1) - k_2 2^{\frac{1+q_1}{2}} V_i^{\frac{1+\beta}{2}}(\tau_1). \end{aligned} \tag{12}$$

According to the fixed-time stability criterion [20] (Lemma 1), there exists $\tau_1^\dagger > 0$ such that $\lim_{\tau_1 \rightarrow \tau_1^\dagger -} \bar{s}_i(\tau_1) = \mathbf{0}_n$ and $\bar{s}_i(\tau_1) \equiv \mathbf{0}_n, \forall \tau_1 \geq \tau_1^\dagger$. Then it follows from (10) and (11) that $\lim_{t \rightarrow t_0 + T_1^\dagger} s_i(t) = \lim_{\tau_1 \rightarrow \tau_1^\dagger} \bar{s}_i(\tau_1) = \mathbf{0}_n$ and $s_i(t) \equiv \mathbf{0}_n, \forall t \geq t_0 + T_1^\dagger$ where $T_1^\dagger = T_1 \tanh(\tau_1^\dagger) < T_1$. Given that $\rho_1(t)$ is bounded on $[t_0, t_0 + T_1^\dagger]$ and removed from the input when $t > t_0 + T_1^\dagger$, $v_i^s(t)$ remains free of singularity. Since $s_i(t) \equiv \mathbf{0}_n$ and $\sum_{i=1}^N r_i(t) - \sum_{i=1}^N b_i = \sum_{i=1}^N s_i(t), \forall t \in [t_0 + T_1, +\infty)$, there is $\sum_{i=1}^N r_i(t) - \sum_{i=1}^N b_i \equiv \mathbf{0}_n, \forall t \in [t_0 + T_1, +\infty)$. Furthermore, system (6) with protocol (7) becomes

$$\dot{r}_i(t) = v_i^f(t) = \begin{cases} -\rho_2(t) \mathcal{F}_{p_2,q_2}(r_i(t), \mathbf{r}_{\mathcal{N}_i}(t)), & \text{if } \tilde{c}_i(t) \neq 1, \\ -\mathcal{F}_{p_2,q_2}(r_i(t), \mathbf{r}_{\mathcal{N}_i}(t)), & \text{otherwise.} \end{cases} \tag{13}$$

Stage II: Letting

$$\tau_2(t; T_2) = \frac{1}{2} \ln \frac{T_2 + t - t_1}{T_2 - (t - t_1)} : [t_1, t_1 + T_2) \rightarrow \mathbb{R}_{\geq 0}, \tag{14}$$

one has $\tau_2(t; T_2) = \mu_2(t)$ and $\mu_2(t) = \frac{(e^{\tau_2} + e^{-\tau_2})^2}{4T_2}$. Moreover, since (14) is invertible, its inverse can be shown as follows,

$$t(\tau_2; T_2) = t_1 + T_2 \tanh(\tau_2) : \mathbb{R}_{\geq 0} \rightarrow [t_1, t_1 + T_2). \tag{15}$$

To implement the time space deformation in (7), the following state transformations are applied,

$$\bar{r}_i(\tau_2) = r_i(t(\tau_2; T_2)). \tag{16}$$

By differentiating (16) along the solutions of (13) when $t \leq t_1 + T_2$ and $\tilde{c}_i(t) \neq 1$, the following scaled model is derived using (15),

$$\begin{aligned} \dot{\bar{r}}_i(\tau_2) &= \frac{d\bar{r}_i(t(\tau_2;T_2))}{dt(\tau_2;T_2)} \frac{\partial t(\tau_2;T_2)}{\partial \tau_2} \\ &= -\mathcal{F}_{p_2,q_2}(\bar{r}_i(\tau_2), \bar{\mathbf{r}}_{\mathcal{N}_i}(\tau_2)) \\ &= -c_1 \sum_{j=1}^N a_{ij} \text{sig}^{p_2}(\nabla f_i(\bar{r}_i(\tau_2)) - \nabla f_j(\bar{r}_j(\tau_2))) \\ &\quad -c_2 \sum_{j=1}^N a_{ij} \text{sig}^{q_2}(\nabla f_i(\bar{r}_i(\tau_2)) - \nabla f_j(\bar{r}_j(\tau_2))) \end{aligned} \tag{17}$$

According to [21] (Theorem 1), there exists $\tau_2^\dagger > 0$ such that $\lim_{\tau_2 \rightarrow \tau_2^\dagger} \bar{r}_i(\tau_2) = x_i^*$ and $\bar{r}_i(\tau_2) \equiv x_i^*, \forall \tau_2 \geq \tau_2^\dagger$.

Then it follows from (16) and (17) that $\lim_{t \rightarrow t_1 + T_2^\dagger} r_i(t) = \lim_{\tau_2 \rightarrow \tau_2^\dagger} \bar{r}_i(\tau_2) = x_i^*$ and $r_i(t) \equiv x_i^*, \forall t \geq t_1 + T_2^\dagger$ where $T_2^\dagger = T_2 \tanh(\tau_2^\dagger) < T_2$. Given that $\rho_2(t)$ is bounded on $[t_1, t_1 + T_2^\dagger]$ and removed from the input for $t > t_1 + T_2^\dagger$, $u_i^c(t)$ remains free of singularity. Therefore, it can be concluded that the trajectories of all agents in the virtual multi-agent system converge to the optimal solution of problem (5) within the prescribed-time interval $T_1 + T_2$ and the designed virtual input $v_i(t)$ remains free of singularity. \square

Remark 4. Theorem 1 explains why the designed algorithm (7) can achieve singularity-free prescribed-time convergence. Unlike existing literature that relies on time-varying high-gain functions such as $\frac{T}{T-(t-t_0)}$, this work extends the approach to $\frac{T}{T^2-(t-t_0)^2}$ and provides theoretical proof from the perspective of time-domain compression transformation. This enriches the design forms of prescribed-time optimization algorithms.

3.2. Prescribed-Time Signal Tracking Controller: A Singularity-Free Design

Singularity-free prescribed-time tracking controllers are designed for the agents of original second-order multi-agent system (4) such that the agents' outputs $h_i(t) = x_i(t), \forall i \in \mathcal{V}$ track the local optimal reference outputs $h_i^r(t) = r_i$ generated from the optimal signal generator (7).

For multi-agent system (4), denote $e_{i,1} = x_i - r_i, e_{i,2} = y_i, i \in \mathcal{V}$. Then the following tracking error system is obtained from system (4),

$$\begin{cases} \dot{e}_{i,1}(t) = e_{i,2}(t) - \dot{r}_i(t), & i \in \mathcal{V}. \\ \dot{e}_{i,2}(t) = u_i(t), \end{cases} \tag{18}$$

Define $\mu_3(t) = \frac{T_3}{T_3+t_2-t}, \varrho_3(t) := \varrho_3(t; t_2, T_3) = \begin{cases} \mu_3(t), & t \in [t_2, t_2 + T_3), \\ 0, & \text{otherwise,} \end{cases}$, and $t_2 = t_1 + T_2$. Based on system (4), the singularity-free prescribed-time tracking protocol is designed as

$$u_i(t) = \begin{cases} \frac{\varrho_3(t)}{T_3} e_{i,2}(t) - \beta_1 \varrho_3^2(t) \text{sig}^{\alpha_1}(e_{i,1}(t)) - \beta_2 \varrho_3^{2-\alpha_2}(t) \text{sig}^{\alpha_2}(e_{i,2}(t)), & \text{if } e_{i,1}(t) \neq \mathbf{0}_n, \\ -\beta_1 \text{sig}^{\alpha_1}(e_{i,1}(t)) - \beta_2 \text{sig}^{\alpha_2}(e_{i,2}(t)) & , \text{otherwise,} \end{cases} \tag{19}$$

where $\alpha_1 = \frac{\alpha_2 \alpha_3}{2\alpha_3 - \alpha_2}, \alpha_3 = 1, \alpha_2 = \eta \in (1 - \varepsilon_i, 1), \varepsilon_i \in (0, 1), \beta_1, \beta_2 > 0$.

Theorem 2. *If Assumptions 1 and 2 hold, under protocols (19) and (7), the outputs $h_i(t) = x_i(t)$ of all agents converge to x^* of problem (5) within the prescribed-time interval $T_1 + T_2 + T_3$, i.e., $\lim_{t \rightarrow (t_0 + T_1 + T_2 + T_3)^-} x_i(t) = x^*$ and $x_i(t) \equiv x^*, \forall t \in [t_0 + T_1 + T_2 + T_3, +\infty), \forall i \in \mathcal{V}$.*

Proof. The proof's approach relies on the idea of time space deformation. When $t \geq t_2$, by Theorem 1, $r_i(t) \equiv x^*$, and $\dot{r}_i(t) \equiv \mathbf{0}_n$. Letting

$$\tau_3(t) := \tau_3(t; t_2, T_3) = -T_3 \ln \frac{T_3 + t_2 - t}{T_3} : [t_2, t_2 + T_3) \rightarrow \mathbb{R}_{\geq 0}, \tag{20}$$

one has $\dot{\tau}_3(t) = \mu(t)$ and $\mu(t) = e^{\frac{\tau_3(t)}{T_3}}$. Moreover, since (20) is invertible, its inverse can be shown as follows,

$$t(\tau_3) := t(\tau_3; t_2, T_3) = t_2 + T_3(1 - e^{-\frac{\tau_3}{T_3}}) : \mathbb{R}_{\geq 0} \rightarrow [t_2, t_2 + T_3). \tag{21}$$

To implement the time space deformation in (18), the following state transformations are applied,

$$\bar{e}_{i,1}(\tau_3) = e_{i,1}(t(\tau_3)), \tag{22a}$$

$$\bar{e}_{i,2}(\tau_3) = \mu^{-1}(t(\tau_3))e_{i,2}(t(\tau_3)). \tag{22b}$$

By differentiating (22) along the solutions of (18) and (19) under the condition $e_{i,1}(t) \neq \mathbf{0}_n$, the following scaled model is derived using (21),

$$\begin{aligned} \dot{\bar{e}}_{i,1}(\tau_3) &= \frac{de_{i,1}(t(\tau_3))}{dt(\tau_3)} \frac{\partial t(\tau_3)}{\partial \tau_3} \\ &= e_{i,2}(t(\tau_3))e^{-\frac{\tau_3}{T_3}} = \bar{e}_{i,2}(\tau_3), \\ \dot{\bar{e}}_{i,2}(\tau_3) &= \frac{d\mu^{-1}(t(\tau_3))e_{i,2}(t(\tau_3))}{dt(\tau_3)} \frac{\partial t(\tau_3)}{\partial \tau_3} \\ &= \left(-\frac{1}{T_3}e_{i,2}(t(\tau_3)) + \mu^{-1}(t(\tau_3))u_i(t(\tau_3))\right)e^{-\frac{\tau_3}{T_3}} \\ &= -\beta_1 \text{sig}^{\alpha_1}(\bar{e}_{i,1}(\tau_3)) - \beta_2 \text{sig}^{\alpha_2}(\bar{e}_{i,2}(\tau_3)). \end{aligned} \tag{23}$$

According to Lemma 1, there exists $\tau_3^\dagger > 0$ such that $\lim_{\tau_3 \rightarrow \tau_3^\dagger} \bar{e}_{i,1}(\tau_3) = \mathbf{0}_n$ and $\bar{e}_{i,1}(\tau_3) \equiv \mathbf{0}_n, \forall \tau_3 \geq \tau_3^\dagger$. Then it follows from (21) and (22a) that $\lim_{t \rightarrow t_2 + T_3^\dagger} e_{i,1}(t) = \lim_{\tau_3 \rightarrow \tau_3^\dagger} \bar{e}_{i,1}(\tau_3) = \mathbf{0}_n$ and $e_{i,1}(t) \equiv \mathbf{0}_n, \forall t \geq t_2 + T_3^\dagger$ where $T_3^\dagger = T_3(1 - e^{-\frac{\tau_3^\dagger}{T_3}}) < T_3$. Thus, the outputs $h_i = x_i$ of all agents converge to x^* of (5) within the prescribed-time interval $T_1 + T_2 + T_3$, i.e., $\lim_{t \rightarrow (t_0 + T_1 + T_2 + T_3)^-} x_i(t) = x^*$ and $x_i(t) \equiv x^*, \forall t \in [t_0 + T_1 + T_2 + T_3, +\infty), \forall i \in \mathcal{V}$.

Finally, given that $\mu_3(t)$ is bounded on $[t_2, t_2 + T_3^\dagger]$ and removed from the input for $t > t_2 + T_3^\dagger$, $u_i(t)$ remains singularity-free. The proof is now complete. \square

Remark 5. *The motivation behind the prescribed-time tracking protocol (19) is to simplify the complexity of resource allocation in second-order systems through a cascaded design. Specifically, we first consider the following two approaches to solving the resource allocation problem in second-order systems,*

1. Directly designing the input for the second-order system (4) to find the optimal solution of problem (5).
2. First using a virtual first-order system to find the optimal solution, and then controlling the input of system (4) to track the constant signal output by the virtual system.

It can be observed that the latter cascaded design decouples the original task, simplifying the complexity of the convergence proof. Additionally, this design facilitates the future extension of the system to more general cases.

4. Numerical Simulation

This section addresses a practical problem of limited power allocation in an electrical network, aiming to minimize the fatigue damage in individual wind turbines within a wind farm. Such problems often require optimal power allocation within seconds. Let $P_i(t)$ be the real-time power of the i -th turbine (W), P_o the total power to be allocated (W), ω_r^i the low-speed shaft rotational speed of the i -th turbine (rad/s), and V_w^i the equivalent hub wind speed of the i -th turbine (m/s). Then the problem can be formulated as the following optimization problem

$$\begin{aligned} \{P_i^*\}_{i=1}^N &:= \arg \min_{\{P_i\}_{i=1}^N} \sum_{i=1}^N \left[\left(\frac{P_i}{\omega_r^i} \right)^2 + \frac{P_i}{\omega_r^i} + V_w^i{}^2 \right], \\ \text{s.t. } \sum_{i=1}^N P_i &= P_o, \forall i \in \mathcal{V}, \end{aligned} \tag{24}$$

where N is the number of wind turbines in the wind farm. In the subsequent simulations, the case of $N = 6$ is considered, and the data communication network topology among the wind turbines is shown in Figure 2.

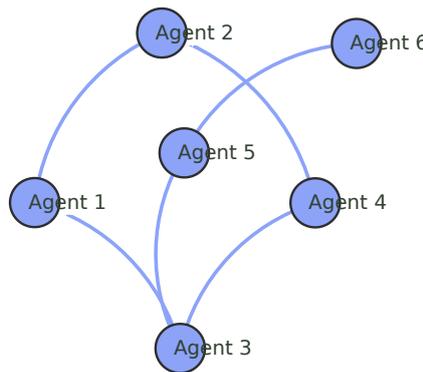


Figure 2. Communication topology.

4.1. Prescribed-Time Resource Allocation in Virtual Wind Farm

For the proposed (7), the prescribed times are set as follows: $T_1 = T_2 = 1$ s. Let $\overline{Re} = \log \left(\sum_{i=1}^5 \frac{\|P_i - P_i^*\|}{\|P_i^*\|} \right)$ denote the residual between the power of each wind turbine and the theoretical optimal power, and $\overline{Er} = \log \left(\sum_{i=1}^N P_i - P_o \right)$ denote the error between the sum of the power of all wind turbines and the actual total power P_o , where $\log(\cdot)$ denotes the natural logarithm. From Figure 3, it can be seen that protocol (7) ensures the output of the virtual system reaches stability within the prescribed time. From Figure 4, it can be observed that the system’s equilibrium point corresponds to the theoretical optimal solution. From Figure 5, it can be seen that the constraint in problem (24) are also satisfied within the prescribed time. In conclusion, it is evident that under protocols (7), all wind turbines can be allocated the optimal power that minimizes the overall fatigue damage within the prescribed time.

Additionally, we compared our work with a related study [17], which proposes a prescribed-time continuous algorithm. To avoid system singularity, their algorithm employs a time-varying scaling function with saturation $\mu_\sigma(t) = \frac{T}{T-t+\sigma}$, where σ is a small positive constant. However, this approach leads to reduced accuracy at the prescribed time. As shown in the convergence error comparison in Figure 6, the accuracy of the algorithm in [17] is lower than that of the algorithm proposed in this paper.

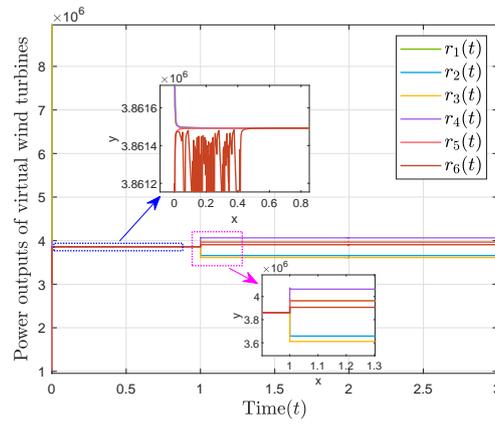


Figure 3. Trajectories of r_i using (7).

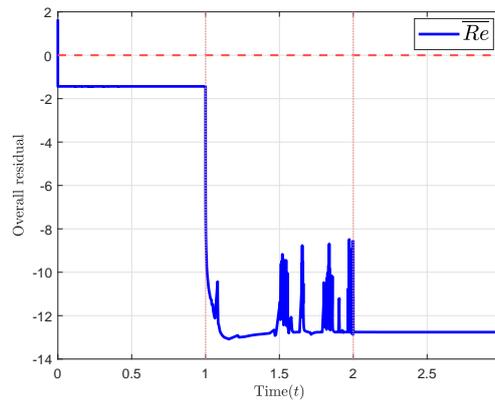


Figure 4. Residual between the power of each wind turbine and the optimal power using (7).

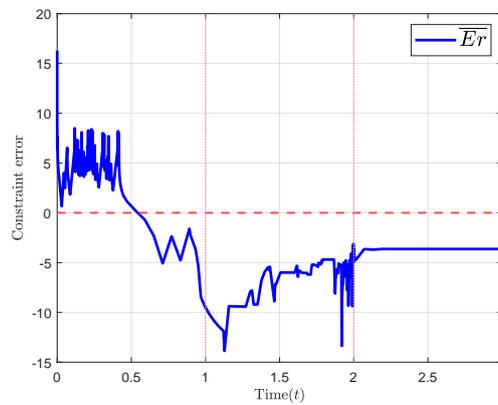


Figure 5. Error between $\sum_{i=1}^N P_i$ and P_o using (7).

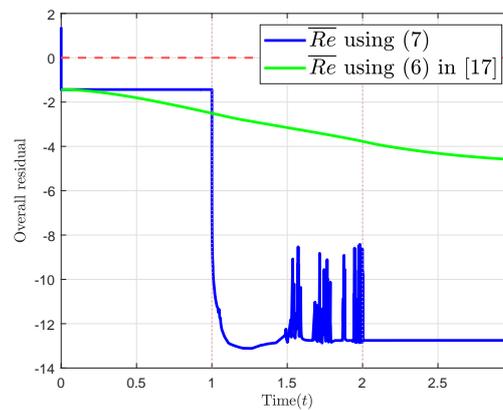


Figure 6. Residual comparison between the proposed (7) and the algorithm in [17].

4.2. Prescribed-Time Signal Tracking

Set the signal tracking time T_3 to 1 s. Using protocol (19), as can be seen from Figure 7, all wind turbines can track the optimal power allocation found in the virtual system within the prescribed time.

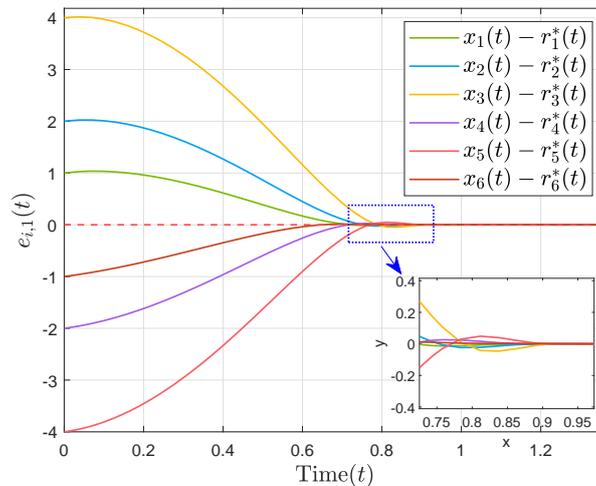


Figure 7. Tracking error using (19).

5. Conclusions

To solve the resource allocation problem in a distributed manner efficiently, this paper develops a singularity-free prescribed-time algorithm. The innovative application of the time-space deformation approach addresses the singularity problem compared to traditional algorithms, enhancing the algorithm's practicality. The proposed singularity-free prescribed-time signal tracking controller for second-order multi-agent systems expands the algorithm's applicability scope. In the future, how to reduce the complexity of the proposed algorithm and expand the generality of the communication topology will become the next research goals.

Author Contributions

S.Z.: methodology, writing—original draft preparation; P.Y.: visualization, data curation; S.Y.: supervision, validation; Y.W.: conceptualization, software, investigation. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (62273092, 61833005), the National Key R&D Project of China (2020YFA0714300), the Jiangsu Provincial Scientific Research Center of Applied Mathematics (BK20233002), the Fundamental Research Funds for the Central Universities (4207012301, 2025SMECP07), and the Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence (2023B1212010001).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Yang, T.; Yi, X.L.; Wu, J.F.; et al. A survey of distributed optimization. *Annu. Rev. Control* **2019**, *47*, 278–305.
2. Liang, S.; Zeng, X.L.; Hong, Y.G. Distributed sub-optimal resource allocation over weight-balanced graph via singular perturbation. *Automatica* **2018**, *95*, 222–228.
3. Huang, J.Y.; Zhou, S.Y.; Tu, H.; et al. Distributed Optimization Algorithm for Multi-Robot Formation with Virtual Reference Center. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 732–734.
4. Yang, S.F.; Liu, Q.S.; Wang, J. Distributed optimization based on a multiagent system in the presence of communication delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 717–728.
5. He, X.; Huang, T.W.; Yu, J.Z.; et al. A continuous-time algorithm for distributed optimization based on multiagent networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *49*, 2700–2709.
6. Wang, B.; Fei, Q.; Wu, Q.H. Distributed time-varying resource allocation optimization based on finite-time consensus approach. *IEEE Control Syst. Lett.* **2020**, *5*, 599–604.
7. Chen, G.; Guo, Z.J. Initialization-free distributed fixed-time convergent algorithms for optimal resource allocation. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 845–854.
8. Dai, H.; Jia, J.P.; Yan, L.; et al. Distributed fixed-time optimization in economic dispatch over directed networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3011–3019.
9. Wang, Y.J.; Song, Y.D.; Hill, D.J.; et al. Prescribed-time consensus and containment control of networked multiagent systems. *IEEE Trans. Cybern.* **2018**, *49*, 1138–1147.
10. Lin, W.T.; Wang, Y.W.; Li, C.J.; et al. Predefined-time optimization for distributed resource allocation. *J. Frankl. Inst.* **2020**, *357*, 11323–11348.
11. Guo, Z.J.; Chen, G. Predefined-time distributed optimal allocation of resources: A time-base generator scheme. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 438–447.
12. Zhang, Y.; Wang, Y.W.; Xiao, J.W.; et al. Distributed predefined-time optimal economic dispatch for microgrids. *Automatica* **2024**, *169*, 111870.
13. Ye, H.F.; Wen, C.Y.; Song, Y.D. Decentralized prescribed-time control for interconnected nonlinear systems via output-feedback. *Automatica* **2024**, *163*, 111571.
14. Ding, Y.; Zhou, B.; Zhang, K.K.; et al. Strong prescribed-time stabilization of uncertain nonlinear systems by periodic delayed feedback. *IEEE Trans. Autom. Control* **2024**, *69*, 4072–4079.
15. Orlov, Y. Time space deformation approach to prescribed-time stabilization: Synergy of time-varying and non-Lipschitz feedback designs. *Automatica* **2022**, *144*, 110485.
16. Zhang, K.P.; Xu, L.; Yi, X.L.; et al. Predefined-time distributed multiobjective optimization for network resource allocation. *Sci. China Inf. Sci.* **2023**, *66*, 170204.
17. Wang, X.Y.; Su, C.X.; Dai, H.; et al. Predefined-time distributed optimization algorithms for a class of resource allocation problem. *J. Frankl. Inst.* **2024**, *361*, 107009.
18. Yue, Y.Y.; Liu, Q.S. Distributed Predefined-Time Convergent Algorithm for Solving Time-Varying Resource Allocation Problem Over Directed Networks. *IEEE Trans. Cybern.* **2025**. <https://doi.org/10.1109/TCYB.2025.3545897>.
19. Bhat, S.P.; Bernstein, D.S. Geometric homogeneity with applications to finite-time stability. *Math. Control Signals Syst.* **2005**, *17*, 101–127.
20. Polyakov, A. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Trans. Autom. Control* **2011**, *57*, 2106–2110.
21. Shi, X.S.; Xu, L.; Yang, T.; et al. Distributed fixed-time resource allocation algorithm for the general linear multi-agent systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 2867–2871.