Data-Driven Model Free Adaptive Sliding Mode Control For Multi DC Motors Speed Regulation

Tony Blaise Bimenyimana

Abstract

Here goes an Abstract

I. INTRODUCTION

IN short time ago, the field of control systems experienced significant advancements, particularly in the domain of multi-agent systems. Distributed coordination of multi-agent systems has been utilized in range of practical applications, including satellite formation, autonomous, underwater vehicules,automated highway systems, and mobile robots. These applications highlight the importance and adaptability of multi-agent systems in addressing complex real world challenges.

A critical aspect of multi-agent systems is robust speed regulation, effectively managing multi direct current (DC) motors is essential for acheiving complex machine movements. The complexity of such systems arises from the necessity to maintain synchronization and optimal performance despite parametric uncertainties and external disturbances. Conventional control methodologies, such us Proportional-Integral-Derivative (PID) control and Linear Quadratic Regulator (LQR), often fall short in these dynamic and unpredictable environments, highlighting the need for more adaptive and resilient approaches.

Firstly, this paper aims to address these challenges by representing an innovative control strategy. Moreover, the proposed method synergizes the strengths of Model Free Adaptive Control (MFAC) and Sliding Mode Control (SMC) to achieve robust and efficient speed regulation in multi-agents systems. Secondly, By leveraging the adaptability of MFAC and the robustness of SMC, this approach seeks to enhance the performance and realibilty of multi DC motors speed regulation in the presence of uncertainties and perturbations.

Model Free Adaptive Control (MFAC) is a technique that eschews the need for an explicit mathematical model of the system, making it highly adaptive to real time changes and disturbances. This adaptability ensures optimal control performance in varying conditions. On the other hand, Sliding Mode Control (SMC) is well known for its robustness and ability to handle Nonlinearities and uncertainties by enforcing high frequency switching control law. SMC ensures that the system states are driven to and maintained within a specified sliding surface.

Finally, the integration of these two methodologies facilitates a control strategy that is both adaptive and resilient. MFAC continiously adjusts the control parameters based on real-time system feedback, while SMC provides a robust framework to manage system uncertainties and external disturbances. This combination is particularly well-suited for the complex dynamics and interactions inherent in multi-agent systems, offering a comprehensive solution to the challenges of robust speed control.

The following sections will outline the remaining content of this paper: section 2 provides——. Section 3 ——————. Section 4 presents simulation results and performance analysis , demonstrating the efficacy of the proposed method under various operating conditions. At the end, Section 5 concludes the paper, summarizing the key findings potential points for future research, including the extension of the proposed approach to other types of multi-agent systems and exploring further enhancements to the control algorithms.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Preliminaries

The set of real numbers is denoted by $\mathbb{R}$. For a given matrix $A \in \mathbb{R}^{n \times n}$, $\|A\|$ represents its matrix norm. The notation $\text{diag}(\cdot)$ refers to a diagonal matrix, and I signifies an identity matrix of appropriate dimensions. In the context of multi-agent systems, graph theory serves as a powerful tool to model interaction topologies. We will now provide a brief introduction to directed graphs within algebraic graph theory. Let $G = (V, E, A)$ be a weighted directed graph, where $V = \{1, 2, \ldots, N\}$ represents the set of vertices, $E \subseteq V \times V$ denotes the set of edges, and A is the adjacency matrix. Here, V also indexes the agents. If agent j can receive a message from agent i, then $(i, j) \in E$, making j the child of i and i the parent of j. The neighborhood of agent i is given by $N_i = \{j \in V \mid (j, i) \in E\}$.

The weighted adjacency matrix $A = (a_{i,j}) \in \mathbb{R}^{N \times N}$ is defined such that $a_{i,i} = 0$, $a_{i,j} = 1$ if $(j, i) \in E$; otherwise, $a_{i,j} = 0$. The Laplacian matrix of G is defined as $L = D - A$, where $D = \text{diag}(d_1^{\text{in}}, d_2^{\text{in}}, \ldots, d_N^{\text{in}})$ and $d_i^{\text{in}} = \sum_{j=1}^{N} a_{i,j}$ is called the in-degree of vertex i. A graph is said to be strongly connected if there exists a path between any pair of vertices.

B. Problem Formulation

In this research, the speed regulation problem of DC motors is frequently examined under the assumption that all motors demonstrate identical dynamic characteristics. Nevertheless, heterogeneity is still a fundamental characteristic of systems comprising multi DC motors. Even if the motors are of the same type and have similar structural features, their parameters can never be exactly the same. This inherent variability makes acheiving coordinated speed regulation across heterogeneous motors a more complex task. Let consider a multi DC motors system consisting of N motors where the interaction topology is represented by G. Assume that each motor i the following nonlinear dynamics is considered:

$$y_i(k + 1) = f_i(y_i(k), u_i(k)), \quad i = 1, 2, \ldots, N \tag{1}$$

Where $y_i(k) \in \mathbb{R}$ represents the output (speed of DC motor), $u_i(k) \in \mathbb{R}$ is the control input (the voltage) and $f_i(\cdot)$ is an unknown nonlinear function, respectively.

We consider a scenario where multiple agents aim to track a consesus trajectory $yd(k)$, which is exclusively accessible to a subset of agents. This trsajectory is assumed to be generated by a virtual leader designated as vertex 0. To model this interaction, we construct a direct graph $G' = (V \cup \{0\}, E', A')$ where V denotes the set of agents, E' represents the edge set defining connections from agents to the virtual leader, and A' constitute a weighted adjacency matrix detailing these connections.

This analysis prosupposes nonlinear dynamics governing each agent's evolution. encompassing dependencies on their respective states and inputs. These nonlinear dynamics accomodate various complexities, facilitating an investigation into how agents align with the desired trajectory $yd(k)$.

Assumption 1: The partial derivative of the nonlinear function $f_i(\cdot)$ with respect to $u_i(k)$ is continuous.

Assumption 2: The model $y_i(k+1) = f_i(y_i(k), u_i(k))$ is generalized Lipschitz, meaning that if $\Delta u_i(k) = u_i(k) - u_i(k-1) \neq 0$, then $|\Delta y_i(k + 1)| \leq b|\Delta u_i(k)|$ holds for any k, where $\Delta y_i(k + 1) = y_i(k + 1) - y_i(k)$ and b is a positive constant.

Remark 1: The practical applicability of the aforementioned assumptions to nonlinear systems has been extensively discussed in [**?**], affirming their suitability for practical multi-agent systems. **Assumption 1** establishes a foundational criterion for controller design, ensuring the continuity of the partial derivative of the nonlinear function $f_i(\cdot)$ with respect to $u_i(k)$. **Assumption 2** implies that the rate of change in an agent's output in response to changes in its control input is bounded. This constraint ensures that, from an energetic perspective, finite changes in control input energy correspond to bounded changes in output energy rates, a crucial consideration for system stability and performance.

C. Linearization Technique

This paper delves into the Compact Form Dynamic Linearization (CFDL) technique, this method simplifies the dynamic of nonlinear systems into a linear form that is easier to handle and control. The CFDL is particularly useful when the control input $u_i(k) = 0$ holds, enabling the system to be described through a compact dynamic linearization model. The system under consideration is governed by the following equation:

$$\Delta y_i(k+1) = \phi_i(k)\Delta u_i(k) \tag{2}$$

Where $|\phi_i(k)| \leq b$. b is a positive constant and the variable $phi_i(k)$ named pseudo-partial-derivative.

We define the distributed measurement output of $xi_i(k)$ for i-th agents as follow:

$$\xi_i(k) = \sum_{j \in N_i} a_{i,j}\left(y_j(k) - y_i(k)\right) + d_i(yd(k) - y_i(k)) \tag{3}$$

In this equation $xi_i(k)$ represents the distributed measurement output of the i-th agent at time step k , N_i denotes the set of the neighboring agents of the i-th agent and $a_{i,j}$ are elemnents of the adjancecy matrix representing the weights between agents i and j. For this equation, we assume that $d_i = 0$, meaning the agent i does not directly consider the disired trajectory $yd(k)$ in the distributed measurement output. However if agent i can receive the disired trajectory ,setting d_i to a non-zero value allows it to align its output with the desired trajectory , improving the control performance.

Let $e_i(k) = y_d(k) - y_i(k)$ denote the tracking error. The objective of this paper is to find an good control law using I/O data of the agents(DC motors), such that the outputs of all agents can track the reference trajectory $y_d(k)$ when only some of agents can access the desired trajectory.

Assumption 3: The communication graph $\bar{G}$ is fixed and strongly connected, with at least one follower agent able to access the leader's trajectory.

Remark 2: This assumption ensures the solvability the tracking problem. An isolated agent, oblivious of the control objective, cannot follow the leader's reference trajectory.

Assumption 4 : The PPD $phi_i(k) > \varsigma, i = 1, 2, 3 \ldots N$ holds for all k, where ς is an rondomly small positive constant. Without loss of generality, in this paper we assume that $phi_i(k)\varsigma$.

Remark 3: This indicatesthat the agent output does not decrease with encreasing control input, resembling a linear characteristic. It implies the control direction is known and unchanging Similar assumptions are common in model-based control and are reasonable for practical systems like mobile robots and UAVs.

III. MAIN RESULTS

Considering the following PPD criterion function:

$$J(\phi_i(k)) = |\Delta y_i(k) - \hat{\phi}_i(k)\Delta u_i(k-1)|^2 + \mu|\hat{\phi}_i(k) - \hat{\phi}_i(k-1)| \tag{4}$$

Differentiating equation (4) with respect to PPD parameter $\phi_i(k)$ and make it equal to zero:

$$\frac{\partial J(\phi_i(k))}{\partial \phi_i(k)} = |\Delta y_i(k) - \hat{\phi}_i(k)\Delta u_i(k-1)|^2 + \mu|\hat{\phi}_i(k) - \hat{\phi}_i(k-1)| = 0 \tag{5}$$

And, let's compute the derivative of $J(\hat{\phi}_i(k))$, we obtain:

$$2[\Delta y_i(k) - \phi_i(k-1)] - [\Delta u_i(k-1)] + 2\mu[\phi_i(k) - \hat{\phi}_i(k-1)] = 0 \tag{6}$$

Then , the following distributed MFAC algorithms is presented:

$$\hat{\phi}_i(k) = \hat{\phi}_i(k-1) + \frac{\eta \Delta u_i(k-1)(\Delta y_i(k) - \hat{\phi}_i(k-1)\Delta u_i(k-1))}{\mu + \Delta u_i(k-1)} \tag{7}$$

$$\hat{\phi}_i(k) = \hat{\phi}_i(1), if |\hat{\phi}_i(k)| \le \epsilon \ or \ sign(\hat{\phi}_i(k)) \ne sign(\hat{\phi}_i(1)) \tag{8}$$

A. Model Free Adaptiive Controller Design

To design the MFAC alogrithm, a performance function $J(u_i(k))$ is set a:

$$J(u_i(k)) = |\xi_i(k)|^2 + \lambda |u_i(k) - u_i(k-1)|^2 \tag{9}$$

This function used to evaluate the effectiveness of the control input $u_i(k)$ for the i-th agents in a control function, with two terms , the first one is the tracking error term $|\xi_i(k)|^2$ where $\xi_i(k)$ represents the distributed measurement output, and by minimizing $|\xi_i(k)|^2$ ensures thet the agent's output closely matches the disired trajectory. The second one is control effect term $\lambda |u_i(k) - u_i(k-1)|^2$ where $u_i(k)$ is the current control input, and $u_i(k-1)$ is the previous control input, and λ is a weighted factor that balances the importance of the control effort.

Subtituting (2) and (3) into (9), then differentiating (9) with respect to $u_i(k)$, and letting it zero, gives:

$$\mathbf{u_{iMFA}}(k) = \mathbf{u_{iMFA}}(k-1) + \frac{\rho \phi_i(k)}{\lambda + |\phi_i(k)|^2} \xi_i(k) \tag{10}$$

Where $\rho \ \varepsilon \ (0,1)$ is a step-size constant, which is added to make (10) general. Using the parameter estimation algorithm (7) and the control law algorithm (10), the MFAC scheme is constructed.

B. Sliding Mode Controller Design

To design the sliding mode control (SMC) for this system, we first define the sliding mode surface, this one guides the system's behavior to ensure robust and accurate tracking of the desired trajectory.

The sliding mode surface is defined as:

$$S_i(k+1) = S_i(k) + e_i(k+1) + \alpha e_i(k) \tag{11}$$

Where $S_i(k)$ represents the sliding surface at the current time step k, $e_i(k)$ is the tracking error , α is a positive constant that influences the dynamics of the sliding surface. The error $e_i(k)$ is defined as the difference between the desired output $y_d(k)$ and the actual output $y_i(k)$.

To ensure that the systm's trajectory is driven toward and remains on the sliding surface, we define a reaching law. The reaching law dictates how quickly the system state converges to the sliding surface and is given by:

$$\Delta S_i(k+1) = -\varepsilon T Sign(k) \tag{12}$$

Where,

$$y_d(k+1) - y_i(k+1) = -\alpha e_i(k) - \varepsilon T Sign(k) \tag{13}$$

In that equation, ε is a small positive constant that controls the rate of the convergence, T is the sampling period, and $Sign(k)$ indicates the direction in which the system should move to reach the sliding surface.

By combining the sliding surface definition and the reaching law, we can derive the control law that ensures the desired tracking performance while maintaining robustness.

The final sliding mode control input $\mathbf{u_{iSM}}(k)$ is designed to be :

$$\mathbf{u_{iSM}}(k) = u_i(k-1) + \frac{y_d(k+1) - y(k) + \alpha e_i(k) + \varepsilon T Sign(k)}{\phi_i(k)} \tag{14}$$

To enhance the robutness and adaptibility of the control system , the Model Free Adaptive Sliding Mode Control (MFASMC) approach is employed. In this approach, the conrol input of the system will be:

$$u_i(k) = \mathbf{u_{iMFA}}(k) + \gamma_i \mathbf{u_{iSM}} \tag{15}$$

Where the parameter γ is a gain factor that adjusts the contribution of the sliding mode control in the control effort and tunes the convergence rate.

C. Stability Analysis

The stability analysis is conducted in two primary steps. The first step focuses on estabilishing the bounds of the error between the actual parameter and its estimated value. The second step ensures that this error remains within acceptable limits over time , leading to a stable system.

Step 1 : The establishment of the bounds of the error between the estimated and actual values of the system's parameter, denote as $\tilde{\phi}_i(k) = \phi_i(k) - \hat{\phi}_i(k)$, starting from the foundational equation derived from the compact dynamic linearization model in equation (2) along with the PPD estimation equation (7), we have:

$$\tilde{\phi}_i(k) = \hat{\phi}_i(k+1) + \frac{\eta \Delta u_i(k-1)}{\mu + |\Delta u_i(k-1)|^2} * ((\Delta y_i(k) - \hat{\phi}_i(k-1)\Delta u_i(k-1))) - \phi_i(k) \tag{16}$$

$$\phi_i(k) = \phi_i(k-1) + (\phi_i(k) - \phi_i(k-1)) \tag{17}$$

$$\tilde{\phi}_i(k) = (\hat{\phi}_i(k+1) - \phi_i(k-1)) + \frac{\eta \Delta u_i(k-1)}{\mu + |\Delta u_i(k-1)|^2} * ((\Delta y_i(k) - \hat{\phi}_i(k-1)\Delta u_i(k-1))) - (\phi_i(k) - \phi_i(k-1)) \tag{18}$$

$$\beta_i(k) = \frac{\eta \Delta u_i(k-1)}{\mu + |\Delta u_i(k-1)|^2} \tag{19}$$

Using equation (19) in equation (18), we get:

$$\tilde{\phi}_i(k) = \tilde{\phi}_i(k-1) + \beta_i(k) * (\Delta y_i(k) - \hat{\phi}_i(k-1)\Delta u_i(k-1) - \phi_i(k) - \phi_i(k-1)) \tag{20}$$

$$\tilde{\phi}_i(k) = (\frac{-}{3}) \tag{21}$$

IV. Dc Motor Speed Mechanism