Consensus protocols for networked multi-agent systems with a uniformly continuous quasi-resetting architecture

Tansel Yucelen\textsuperscript{a} & Wassim M. Haddad\textsuperscript{b}
\textsuperscript{a} Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA
\textsuperscript{b} School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Accepted author version posted online: 16 Jan 2014. Published online: 06 Mar 2014.

\textbf{To cite this article:} Tansel Yucelen & Wassim M. Haddad (2014) Consensus protocols for networked multi-agent systems with a uniformly continuous quasi-resetting architecture, International Journal of Control, 87:8, 1716-1727, DOI: 10.1080/00207179.2014.883647

\textbf{To link to this article:} http://dx.doi.org/10.1080/00207179.2014.883647

\section*{PLEASE SCROLL DOWN FOR ARTICLE}

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions
Consensus protocols for networked multi-agent systems with a uniformly continuous quasi-resetting architecture

Tansel Yucelen\textsuperscript{a,∗} and Wassim M. Haddad\textsuperscript{b}

\textsuperscript{a}Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO, USA; \textsuperscript{b}School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

(Received 21 April 2013; accepted 12 January 2014)

The consensus problem appears frequently in coordination of multi-agent systems in aerospace science and engineering, and involves the agreement of networked agents upon certain quantities of interest. In this paper, we focus on a new consensus protocol for networked multi-agent systems using a resetting control architecture. Specifically, the control protocol consists of a delayed feedback, quasi-resetting control law such that controller resettings occur when the relative state measurements (i.e., distance) between an agent and its neighbouring agents approach zero. In contrast to standard impulsive resetting controllers, the proposed resetting is uniformly continuous, and hence, our approach does not require any well-posedness assumptions imposed by impulsive resetting controllers. In addition, using a Lyapunov–Krasovskii functional, it is shown that the multi-agent system reaches asymptotic state equipartitioning, where the system steady state is uniformly distributed over the system initial conditions. Finally, we develop $L_\infty$ transient performance guarantees while accounting for system overshoot and excessive control effort.

Keywords: networked multi-agent systems; consensus protocols; resetting control; delayed feedback control; quasi-resetting control; transient performance guarantees

1. Introduction

Networked multi-agent systems consist of a group of agents that locally sense their environment, communicate with each other, and process information in order to achieve a given set of system objectives. Since these systems have widespread applications in physics, biology, social sciences, economics, and engineering, it is not surprising that the last decade has witnessed an increased interest in networked multi-agent systems (see, for example, Mesbahi & Egerstedt, 2010; Olfati-Saber, Fax, & Murray, 2007; Ren, Beard, & Atkins, 2007, and references therein). For a multi-vehicle aerospace network of interconnected systems, it is often desired that some property of each vehicle approaches a single common value across the network. For example, in a group of autonomous aerospace vehicles, this property might be a common heading angle or a shared communication frequency. Designing a controller that ensures a set of system objectives is called the consensus control problem (Mesbahi & Egerstedt, 2010; Olfati-Saber et al., 2007; Ren et al., 2007). Consensus control protocols employ a distributed controller architecture wherein local information is accessed and processed.

In this paper, we present a novel network consensus control protocol using an approximate resetting architecture. The notion of resetting in feedback control design was originally introduced in Clegg (1958). Specifically, a non-linear integral controller architecture was proposed, where the integrator resets its output to zero whenever its input is zero. Since resetting controllers are non-linear, closed-loop system performance is not constrained to the Bode integral limitation theorem (Horowitz, 1963), and hence, can be used to overcome fundamental performance limitations of linear controllers for achieving fast system response without excessive overshoot and control effort (Beker, Hollot, & Chait, 2001; Beker, Hollot, Chait, & Han, 2004; Nesić, Zaccrian, & Teel, 2008; Guo, Wang, Xie, & Zheng, 2009).

The contents of the paper are as follows. In Section 2, we establish definitions, notation, and review some basic results from graph theory which provide the mathematical foundation for designing quasi-resetting controllers for multi-agent systems. In Section 3, we highlight our proposed approximate resetting controller approach using a simple first-order integrator model. In Section 4, we generalise the ideas presented in Section 3 to develop continuous approximate resetting controllers for networked multi-agent systems. This controller framework leads to a new consensus protocol architecture consisting of a delayed feedback control law with uniformly continuous quasi-resettings occurring when the relative state measurements (i.e., distance) between an agent and its neighbouring agents approach...
zero. Furthermore, we show that the proposed framework does not require any well-posedness assumptions nor time regularisation that is typically imposed by hybrid resetting controllers. We then turn our attention to stability and convergence in Section 5. Specifically, using a Lyapunov–Krasovskii functional, we show that the multi-agent system reaches asymptotic agreement, wherein the system steady state is uniformly distributed over the system initial conditions preserving the centroid of the network. In addition, we develop $\mathcal{L}_\infty$ consensus performance guarantees while accounting for system overshoot constraints and excessive control effort. In Section 6, we present several illustrative numerical examples to demonstrate the efficacy of the proposed control protocol approach. Finally, we draw conclusions in Section 7.

2. Notation, definitions, and graph-theoretic notions

In this section, we establish notation, definitions, and recall some basic results from graph theory (Mesbahi & Egerstedt, 2010; Godsil & Royle, 2001). Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n \times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+$ denotes the set of positive real numbers, $\mathbb{N}$ (respectively, $\mathbb{P}$) denotes the set of $n \times n$ non-negative-definite (respectively, positive-definite) real matrices, $\mathbb{S}^n$ denotes the set of $n \times n$ symmetric real matrices, $0_n$ denotes the $n \times 1$ zero vector, $e_n$ denotes the $n \times 1$ ones vector, that is, $e_n = [1, \ldots, 1]^T \in \mathbb{R}^n$, $0_n \times n$ is the $n \times n$ zero matrix, and $I_n$ denotes the $n \times n$ identity matrix. In addition, we write $(\cdot)^T$ for the transpose operator, $(\cdot)^{-1}$ for the inverse operator, $\otimes$ for the Kronecker product (Bernstein, 2009, p. 440), $\|\cdot\|_2$ for the Euclidian vector norm, $\lambda_{\text{min}}(A)$ (respectively, $\lambda_{\text{max}}(A)$) for the minimum (respectively, maximum) eigenvalue of the Hermitian matrix $A$, $\lambda_i(A)$ for the $i$th eigenvalue of the square matrix $A$, $\text{diag}(a)$ for the diagonal matrix with the entries $a \in \mathbb{R}^n$ on its diagonal, and $[A]_{ij}$ for the $(i, j)$th entry of the matrix $A$. Finally, for a signal $x(t) \in \mathbb{R}^n$, $t \geq 0$, the extended $\mathcal{L}_\infty$ norm and the $\mathcal{L}_\infty$ norm (Khalil, 2002, Section 5) are defined, respectively, as $\|x\|_{\mathcal{L}_\infty} \triangleq \max_{1 \leq t \leq T} (\sup_{0 \leq s \leq t} |x(s)|)$ and $\|x\|_{\mathcal{L}_\infty} \triangleq \max_{1 \leq t \leq n} (\sup_{0 \leq s \leq t} |x(t)|)$, where $x(t)$ denotes the $i$th component of $x(t)$ and $T > 0$.

In the networked multi-agent system literature, graphs are broadly adopted to encode interactions between groups of agents. An undirected graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1, \ldots, n\}$ of nodes and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of edges. If $(i, j) \in \mathcal{E}_\mathcal{G}$, then the nodes $i$ and $j$ are neighbours and the neighbouring relation is indicated by $i \sim j$. The degree of a node is given by the number of its neighbours. If $d_i$ is the degree of node $i$, then the degree matrix of a graph $\mathcal{G}$ denoted by $D(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is given by $D(\mathcal{G}) \triangleq \text{diag}(d)$, where $d = [d_1, \ldots, d_n]^T$.

A path $i_0, i_1, \ldots, i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1, \ldots, L$, and a graph $\mathcal{G}$ is connected, if there exists a path between any pair of distinct nodes. The adjacency matrix of a graph $\mathcal{G}$ denoted by $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is given by

$$
[\mathcal{A}(\mathcal{G})]_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{E}_\mathcal{G}, \\ 0, & \text{otherwise}. \end{cases}
$$

The Laplacian matrix of a graph $\mathcal{L}(\mathcal{G}) \in \mathbb{N}^n \otimes \mathbb{S}^n$ plays a central role in graph theory for networked multi-agent systems and is given by $\mathcal{L}(\mathcal{G}) \triangleq D(\mathcal{G}) - \mathcal{A}(\mathcal{G})$, where the eigenvalues of the Laplacian for a connected undirected graph can be ordered as

$$
0 = \lambda_{\text{min}}(\mathcal{L}(\mathcal{G})) \leq \lambda_2(\mathcal{L}(\mathcal{G})) \leq \cdots \leq \lambda_n(\mathcal{L}(\mathcal{G})).
$$

It can be easily shown that $e_n$ is the eigenvector corresponding to the zero eigenvalue $\lambda_{\text{min}}(\mathcal{L}(\mathcal{G}))$ of $\mathcal{L}(\mathcal{G})$, and hence, $\mathcal{L}(\mathcal{G})e_n = 0_n$ holds.

As an example, consider the networked multi-agent system shown in Figure 1. For this system, the graph Laplacian $\mathcal{L}(\mathcal{G})$ has the form

$$
\mathcal{L}(\mathcal{G}) \triangleq \begin{bmatrix}
2 & -1 & 0 & 0 & 0 & \ldots \\
-1 & 2 & -1 & 0 & 0 & \ldots \\
0 & -1 & 2 & -1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots
\end{bmatrix} \in \mathbb{N}^{12} \otimes \mathbb{S}^{12}.
$$

3. A quasi-resetting control architecture for approximating resetting controllers

In this section, we highlight the salient features of our quasi-resetting control framework for approximating resetting hybrid controllers using a simple first-order integrator model. Specifically, to elucidate our proposed approach, consider the dynamical system given by $\dot{x}(t) = u(t)$, $x(0) = 0$, where $x(t) \in \mathbb{R}$, $t \geq 0$, is the system state and the control law $u(t)$, $t \geq 0$, is given by

$$
u(t) = -k_1(x(t) - c) - v(t),
$$

$$
u(t) = k_2(x(t) - c), \quad v(0) = 0, \quad x(t) - c \neq 0, \quad v(t^+) = 0, \quad x(t) - c = 0.
$$

Figure 1. A networked multi-agent system represented by an undirected graph.
where $c \in \mathbb{R}$ is a constant command, $v(t) \in \mathbb{R}$ is an integrator state, $v(t^+) = \lim_{\varepsilon \to 0} v(t + \varepsilon)$, and $k_i > 0$, $i = 1, 2$, are the design parameters. The tracking performance based on the control law (4)-(6) for $c = 1$ is shown in Figure 2 without and with resetting. In particular, this simple example illustrates the desired effect of the resetting action of the integrator and how it yields satisfactory tracking performance in finite time.

The stability properties of the aforementioned simple example can be shown by considering the Lyapunov function candidate $V(e, v) = e^2 + k_2^{-1}v^2$, where $e(t) \triangleq x(t) - c$. Note that $v(t) = 0$ and $V(e, v) > 0$ for all $(e, v) \neq 0$. Since $\dot{V}(e(t), v(t)) = -2k_1e^2(t) \leq 0$, $(e(t), v(t)) \not\in \mathcal{Z}$, and $\Delta V(e(t), v(t)) \triangleq V(e(t^+)), v(t^+)) - V(e(t), v(t)) = -k_2^{-1}v^2(t) \leq 0$, $(e(t), v(t)) \in \mathcal{Z}$, where $\mathcal{Z} \triangleq \{(e, v) \in \mathbb{R} \times \mathbb{R} : e = 0\}$ is the resetting set, it follows from Theorem 2.3 of Haddad, Chellaboina, and Nersesov (2006) that $(e(t), v(t)) \to \mathcal{M} \triangleq \{(0, 0)\}$ as $t \to \infty$, where $\mathcal{M}$ is the largest invariant set contained in $\mathcal{R} \triangleq \{(e(t), v(t)) \in \mathbb{R} \times \mathbb{R} : (e(t), v(t)) \not\in \mathcal{Z}, \dot{V}(e(t), v(t)) = 0 \} \cup \{(e(t), v(t)) \in \mathbb{R} \times \mathbb{R} : (e(t), v(t)) \in \mathcal{Z}, \Delta V(e(t), v(t)) = 0\}$. In order to apply Theorem 2.3 of Haddad et al. (2006), we require that (1) if $(e(t), v(t)) \in \mathcal{Z}$, then $(e(t^+), v(t^+)) \not\in \mathcal{Z}$, and (2) if at time $t$ the trajectory $(e(t), v(t))$ belongs to the closure of $\mathcal{Z}$ but not $\mathcal{Z}$, then there exists $\epsilon > 0$ such that for all $0 < \delta < \epsilon$, $(e(t + \delta), v(t + \delta)) \not\in \mathcal{Z}$.

Assumptions (1) and (2) guarantee that for a particular system initial condition, the resetting system times are well defined and distinct (Haddad et al., 2006). Even though these assumptions can be satisfied for the preceding example by redefining the resetting set as in, for example, Banos, Carrasco, and Barreiro (2011), for certain problem formulations that may prove to be a difficult task. Furthermore, even if Assumptions (1) and (2) hold, then the closed-loop system with a resetting controller can exhibit Zeno solutions, wherein solutions exhibit infinitely many resettings in finite time (Haddad et al., 2006). To circumvent this problem, a time regularisation approach can be used (Nesic et al., 2008; Banos et al., 2011), wherein resetting is avoided if a minimum dwell time between the resetting times is enforced. However, in general, it is desirable to eliminate such assumptions since they tend to require specialised hybrid controller architectures (Haddad, Chellaboina, Hui, & Nersesov, 2007; Haddad, Nersesov, & Ghasemi, 2013).

In practice, implementation of a resetting controller may not be feasible. To see this, consider the algorithm given in Table 1 for the resetting controllers (5) and (6), where $\epsilon > 0$ is a small constant and $\Delta t$ is the sampling time. Here, we approximate Equation (5) by resorting to a first-order Euler integration method and implement Equation (6) by using if-else logic. Even though $\Delta t \to 0$ results in the exact implementation of Equations (5) and (6), this implementation is not possible in practice.

To address the implementation issues discussed above, consider the delayed feedback control law given by

![Figure 2. Tracking performance for a constant command $c = 1$ using the control law given by Equations (4)-(6) with and without resetting ($k_1 = 0.2$ and $k_2 = 1$).](image)

Table 1. Algorithm for the resetting controller in Equations (5) and (6).

| $\text{if } |x-c| \leq \epsilon; v = 0; \text{ else } v = v + \Delta t[k_3(x-c)]; \text{ end.}$ |
Equation (4) with, in place of Equations (5) and (6),

\[ v(t) = \alpha(x(t) - c)v(t - \tau) + k_3(x(t) - c), \quad (7) \]

where

\[ \alpha(x(t) - c) \triangleq [1 - \text{sech}^2(\xi (x(t) - c))] \theta, \quad (8) \]

\( \xi > 0, \theta > 0, k_3 > 0 \) are design parameters, and \( \tau > 0 \) is a time-delay design parameter (see Figure 3). In the case where \( x(t) - c \) is sufficiently bounded away from zero, it follows from Equation (8) that \( \alpha(x(t) - c) \approx 1 \), and hence, Equation (7) behaves as an integrator since \( v(t) \approx v(t - \tau) + k_3(x(t) - c) \). Alternatively, if \( x(t) - c = 0 \), then \( \alpha(x(t) - c) = 0 \), and hence, \( v(t) = 0 \), so that approximate resetting occurs. In contrast to standard impulsive resetting controllers (Haddad et al., 2007, 2013), this approximate resetting involves a quasi-impulsive uniformly continuous switch. From an implementation perspective, the time-delay design parameter \( \tau \) can be chosen to be sufficiently small so that \( \tau = \Delta t \); see Table 2 for the implementation of Equation (7).

Figure 4 revisits the preceding example and shows the tracking performance for \( c = 1 \) with the delayed feedback control law given by Equations (4) and (7) for two different values of \( \xi \). For both values of \( \xi \), the controller achieves satisfactory tracking performance. Furthermore, note that for \( \xi = 250 \), the proposed approximate resetting controller (7) recovers the tracking performance shown in Figure 2 for the resetting controllers (5) and (6).

<table>
<thead>
<tr>
<th>Table 2. Algorithm for the quasi-resetting controller in Equation (7).</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v = [1 - \text{sech}^2(\xi (x - c))] \theta ) ( v + k_3(x - c) ).</td>
</tr>
</tbody>
</table>

4. Consensus control protocols

In this section, we generalise the ideas presented in Section 3 to develop continuous approximate resetting controllers for networked multi-agent systems. Specifically, we consider a system of \( n \) agents exchanging information using local measurements with \( G \) defining a connected undirected graph topology and with nodes and edges representing agents and inter-agent information exchange links, respectively. Specifically, let \( x_i(t) \in \mathbb{R}^m \) denotes the state of node \( i \) at time \( t \geq 0 \) whose dynamics is described by the single integrator dynamics:

\[ \dot{x}_i(t) = u_i(t), \quad x_i(0) = x_{i0}, \quad i = 1, \ldots, n, \quad (9) \]

where \( u_i(t) \in \mathbb{R}^m, t \geq 0 \), is the control input of node \( i \). Assuming agent \( i \) has access to the relative state information with respect to its neighbours, a standard solution of the consensus problem can be achieved by applying the standard control protocol (Mesbahi & Egerstedt, 2010):

\[ u_i(t) = -k_1 \sum_{i-j} (x_i(t) - x_j(t)), \quad (10) \]

where \( k_1 > 0 \) is a design parameter. Here, Equation (9), in conjunction with Equation (10), can be represented by the graph Laplacian dynamics:

\[ \dot{x}(t) = -k_1 L(G) \otimes I_m x(t), \quad x(0) = x_0, \quad t \geq 0, \quad (11) \]

where \( x(t) = [x_1^T(t), \ldots, x_n^T(t)]^T \in \mathbb{R}^{mn} \) denotes the aggregated state vector of the multi-agent system.

Although our results can be directly extended to the case of Equation (11), for simplicity of exposition, we will focus on individual agent states evolving in \( \mathbb{R} \) (i.e., \( m = 1 \)). In this case, Equation (11) becomes

\[ \dot{x}(t) = -k_1 L(G)x(t), \quad x(0) = x_0, \quad t \geq 0. \quad (12) \]
Since the undirected graph $\mathcal{G}$ is connected, it can be shown that $x(t) \to \frac{1}{n}e_n e_n^T x_0$ as $t \to \infty$ (Mesbahi & Egerstedt, 2010). This shows that agents reach asymptotic agreement to a steady-state value composed of an average of the system initial conditions; that is, centroid convergence of the network.

Next, we generalise the control protocol given by Equation (10) by developing continuous quasi-resetting control protocol for networked multi-agent systems. Specifically, let

$$u_i(t) = -k_1 \sum_{i \neq j} (x_i(t) - x_j(t)) - \sum_{i \neq j} (v_i(t) - v_j(t)), \quad \text{(13)}$$

$$v_i(t) = \alpha_i \left( \sum_{i \neq j} (x_i(t) - x_j(t)) \right) v_i(t - \tau) + k_2 \sum_{i \neq j} (x_i(t) - x_j(t)), \quad \text{(14)}$$

where

$$\alpha_i \left( \sum_{i \neq j} (x_i(t) - x_j(t)) \right) \triangleq \left[ 1 - \text{sech}^2 \left( \xi \sum_{i \neq j} (x_i(t) - x_j(t)) \right) \right]^\theta, \quad \text{(15)}$$

$\xi > 0$, $\theta > 0$, $v_i(t) \in \mathbb{R}$, $k_1 > 0$, and $k_2 > 0$ are design parameters, and $\tau > 0$ is a time-delay design parameter. Note that in the case where $[\mathcal{L}(\mathcal{G})]_{11} \triangleq \sum_{i \neq j} (x_i(t) - x_j(t))$ is sufficiently bounded away from zero, then $\alpha_i(\cdot) \approx 1$, and hence, Equation (14) serves as an integrator, since

$$v_i(t) \approx v_i(t - \tau) + k_2 \sum_{i \neq j} (x_i(t) - x_j(t)). \quad \text{(16)}$$

Alternatively, if $[\mathcal{L}(\mathcal{G})]_1 = 0$, that is, the relative state measurements between an agent $i$ and its neighbouring agents vanish, then $\alpha_i(\cdot) = 0$, and hence, $v_i(t) = 0$, so that resetting occurs.

The proposed uniformly continuous quasi-resetting control protocol (13) can be related to standard impulsive resetting controllers. To see this, let $\xi \to \infty$ and $\theta \to \infty$ in Equation (15). In this case, $\alpha_i(\cdot) \approx 1$ and $\alpha_i(\cdot) = 0$, when $[\mathcal{L}(\mathcal{G})]_1 \neq 0$ and $[\mathcal{L}(\mathcal{G})]_1 = 0$, respectively. For $[\mathcal{L}(\mathcal{G})]_1 \neq 0$ (i.e., $\alpha_i(\cdot) = 1$), it follows from Equation (14) that

$$v_i(t) = v_i(t - \tau) + k_2 \sum_{i \neq j} (x_i(t) - x_j(t)), \quad \text{(17)}$$

or, equivalently,

$$[v_i(t) - v_i(t - \tau)]/\tau = \hat{k}_2 \sum_{i \neq j} (x_i(t) - x_j(t)), \quad \text{(18)}$$

where $\hat{k}_2 \triangleq k_2/\tau$. Now, letting $\tau \to 0$ yields

$$\dot{v}_i(t) = \hat{k}_2 \sum_{i \neq j} (x_i(t) - x_j(t)), \quad \text{(19)}$$

Figure 4. Tracking performance for a constant command $c = 1$ using the delayed control law given by Equations (4) and (7) with $\xi = 25$ and $\xi = 250$ ($\tau = 0.001$, $k_1 = 0.001$, and $\theta = 1$).
where \( \tilde{v}_i(t) = \lim_{\tau \to 0} [v_i(t) - v_i(t - \tau)]/\tau \). In addition, for \( [L(\Theta)x]_i = 0 \) (i.e., \( \alpha_i(\cdot) = 0 \)), Equation (14) can be viewed as \( v_i(t^+) = 0 \) in light of the conditions on \( \xi, \theta \), and \( \tau \). That is, for \( \xi \to \infty, \theta \to \infty \), and \( \tau \to 0 \), the standard impulsive reset controller \( i \) is given by

\[
\dot{v}_i(t) = \hat{k}_2 \sum_{i \sim j} (x_i(t) - x_j(t)), \quad [L(\Theta)x]_i \neq 0, \quad (20)
\]

\[
v(t^+) = 0, \quad [L(\Theta)x]_i = 0. \quad (21)
\]

As noted in Section 3, using Equations (20) and (21) instead of Equations (14) and (15) requires well-posedness assumptions and time regularisation to guarantee that the system resetting times are distinct and well defined, as well as guarantee the absence of Zeno solutions.

5. Stability analysis

In this section, we establish stability properties of the proposed consensus control protocol with quasi-resetting given by Equations (13) and (14). Let \( x(t) = [x_1(t), \ldots, x_n(t)]^T \in \mathbb{R}^n \) and \( v(t) = [v_1(t), \ldots, v_n(t)]^T \in \mathbb{R}^n \). Then, Equation (9), in conjunction with Equations (13) and (14), can be written as

\[
\dot{x}(t) = -k_1 L(\Theta)x(t) - L(\Theta)v(t), \quad x(0) = x_0, \quad t \geq 0, \quad (22)
\]

\[
v(t) = D(\Theta)v(t - \tau) + k_2 L(\Theta)x(t), \quad (23)
\]

where \( D(\Theta) \in \mathbb{P}^n \cap \mathbb{S}^n \) is given by

\[
D(\Theta) \triangleq \text{diag}(\alpha(\cdot)), \quad \alpha(\cdot) = [\alpha_1(\cdot), \ldots, \alpha_n(\cdot)]^T. \quad (24)
\]

Note that for every \( \kappa > 0, I_n - D^2(\Theta) \in \mathbb{P}^n \cap \mathbb{S}^n \).

The following key lemmas are necessary for the main results of this section. For the following results, we consider the transformation given by

\[
x(t) = \text{ave}(x(t)) e_n + \delta(t), \quad (25)
\]

where \( \text{ave}(x(t)) \triangleq \frac{1}{n} e_n^T x(t) \) and \( \delta(t) \in \mathbb{R}^n, t \geq 0 \).

**Lemma 5.1:** Consider the dynamical system given by

\[
\dot{x}(t) = -k_1 L(\Theta)x(t) - L(\Theta)\rho(t), \quad x(0) = x_0, \quad t \geq 0, \quad (26)
\]

where \( \rho(t) \in \mathbb{R}^n, t \geq 0 \), is an arbitrary vector. Then, \( \text{ave}(x(t)) = \frac{1}{n} e_n^T x_0 \) for all \( t \geq 0 \).

**Proof:** Differentiating \( \text{ave}(x(t)) \) with respect to time yields

\[
\frac{d}{dt} \text{ave}(x(t)) = \frac{1}{n} e_n^T x(t) = \frac{1}{n} e_n^T [-k_1 L(\Theta)x(t) - L(\Theta)\rho(t)] = \text{ave}(x(t)), \quad t \geq 0,
\]

where in Equation (27) we used the fact that \( L(\Theta)e_n = 0_n \) and \( L(\Theta) = L(\Theta)^T \). Hence, \( \text{ave}(x(t)) = \text{ave}(x(0)), t \geq 0 \), which proves the result. \( \square \)

**Lemma 5.2:** Consider the dynamical system given by Equation (22) with \( x(t) \) given by Equation (25). If \( \delta(t) \to 0 \) as \( t \to \infty \), then \( x(t) \to \frac{1}{n} e_n e_n^T x_0 \) as \( t \to \infty \).

**Proof:** It follows from Equation (25) that \( \delta(t) \to 0 \) as \( t \to \infty \) implies \( x(t) = \text{ave}(x(t)) e_n \) as \( t \to \infty \). Furthermore, since the form of Equation (22) is identical to Equation (26), it follows from Lemma 5.1 that \( \text{ave}(x(t)) = \frac{1}{n} e_n^T x_0, t \geq 0 \), and hence, \( x(t) \to \frac{1}{n} e_n e_n^T x_0 \) as \( t \to \infty \). \( \square \)

**Remark 5.1:** Note that if \( \lim_{t \to \infty} \delta(t) = 0 \), then asymptotic agreement with the system steady state uniformly distributed over the system initial conditions preserving the centroid of the network is guaranteed.

An alternative consensus protocol using a quasi-resetting architecture can be constructed by considering

\[
u_i(t) = -k_1 \sum_{i \sim j} (x_i(t) - x_j(t)) - v_i(t) \quad (28)
\]

and Equation (14). In this case, we have

\[
\dot{x}(t) = -k_1 L(\Theta)x(t) - v(t), \quad x(0) = x_0, \quad t \geq 0, \quad (29)
\]

and Equation (23). However, since the form of Equation (29) is different than Equation (26), \( \text{ave}(x(t)) \neq \text{ave}(x(0)), t \geq 0 \), that is, centroid convergence of the network is no longer guaranteed at steady state. Note that for \( \alpha(\cdot) \equiv 1 \) and \( \tau \to 0 \), Equations (28) and (14) can be viewed as a delayed version of the proportional–integral protocols proposed in Taylor, Beard, and Humpherys (2011) and Yucelen and Egerstedt (2012). To see this, let \( \tilde{k}_2 \triangleq k_2/\tau \) and note that the integral action

\[
\dot{v}_i(t) = \tilde{k}_2 \sum_{i \sim j} (x_i(t) - x_j(t)) \quad (30)
\]

can be recovered, where \( \dot{v}_i(t) = \lim_{\tau \to 0} [v_i(t) - v_i(t - \tau)]/\tau \).}

**Lemma 5.2** suggests that we need \( \lim_{t \to \infty} \delta(t) = 0 \) when solving the consensus problem. For this purpose, we use the transformation given by Equation (25) to write Equations (22) and (23) as

\[
\delta(t) = -k_1 L(\Theta)\delta(t) - L(\Theta)v(t), \quad \delta(0) = \delta_0, \quad t \geq 0, \quad (31)
\]
\[ v(t) = D(\Theta)v(t - \tau) + k_2 L(\Theta)d(t). \] (32)

The following theorem is necessary for the main results of this paper.

**Theorem 5.1:** Consider the networked multi-agent system given by Equation (9) where agents exchange information using local measurements and with \( \Theta \) defining a connected undirected graph topology. Furthermore, consider the consensus protocol given by Equations (13), (14), and (15). Then, \( \lim_{t \to \infty} \delta(t), v(t) = (0, 0) \).

**Proof:** Consider the Lyapunov–Krasovskii functional candidate given by

\[ V(\delta, v) = \delta^T \delta + k_2^{-1} \int_{-\tau}^{0} v^T(\mu)v(\mu)d\mu \] (33)

and note that \( V(0, 0) = 0 \) and \( V(\delta, v) > 0 \) for all \( \delta, v \neq (0, 0) \). Differentiating \( \delta^T \delta \) along the system trajectory of Equation (31) yields

\[
\frac{d}{dt} \int_{-\tau}^{t} v^T(\mu)v(\mu)d\mu = v^T(t)v(t) - v^T(t - \tau)v(t - \tau)
\]

Next, it follows from Equations (33), (34), and (35) that

\[
\hat{V}(\delta(t), v(t)) \leq -2k_2 \delta^T(t)L(\Theta)d(t)
\]

Now, let \( R = \{ \delta(t), v(t) \in \mathbb{R}^n \times \mathbb{R}^n : \hat{V}(\delta(t), v(t)) = 0 \} \) and let \( M \) be the largest invariant set contained in \( R \). Note that, in this case, since \( L(\Theta)d(t) = 0_n \), it follows that \( \sum_{i=1}^{n} \delta_i(t) - \delta_j(t) = 0 \) for all \( i, j \in V_{\Theta} \). Using similar arguments as in Olfati-Saber and Murray (2003, Theorem 3), it follows from the connectivity of the graph \( \Theta \) that \( \delta_i(t) = \delta_j(t) \) for all \( i, j \in V_{\Theta} \). Furthermore, by Equation (25), \( \delta(t) = (I_n - \frac{1}{n} e_i e_i^T)x(t) \), which implies that \( \sum_{i=1}^{n} \delta_i(t) = 0 \). Finally, in this case, since \( D(\Theta) = 0_{n \times n} \), it follows from Equation (32) that \( v(t) = 0 \). Hence, \( (\delta(t), v(t)) \to M = \{(0, 0)\} \) as \( t \to \infty \).

The following results highlighting asymptotic agreement of the networked multi-agent system and worst-case transient performance guarantees, respectively, are now immediate. For the first result, recall the definition of semi-stability given in Haddad and Chellaboina (2008, p. 261).

**Theorem 5.2:** Consider the networked multi-agent system given by Equation (9) where agents exchange information using local measurements and with \( \Theta \) defining a connected undirected graph topology. Furthermore, consider the consensus protocol given by Equations (13), (14), and (15). Then, \( \lim_{t \to \infty} x(t) = \frac{1}{n} e_i e_i^T x_0 \) and \( \frac{1}{n} e_i e_i^T x_{0} \) is a semi-stable equilibrium state of Equation (9) with \( u(t) \) given by Equation (13).

**Proof:** The proof is a direct consequence of Lemma 5.2 and Theorem 5.1 with the Lyapunov–Krasovskii functional given by \( V(x, v) = x^T x + k_2^{-1} \int_{-\tau}^{0} v^T(\mu)v(\mu)d\mu \).

**Theorem 5.3:** Consider the networked multi-agent system given by Equation (9) where agents exchange information using local measurements and with \( \Theta \) defining a connected undirected graph topology. Furthermore, consider the consensus protocol given by Equations (13), (14), and (15). Then,

\[
\left\| x - \frac{1}{n} e_i e_i^T x_0 \right\|_{\infty} \leq \sqrt{V \left[ \left( I_n - \frac{1}{n} e_i e_i^T \right)x_0, v(0) \right].}
\] (37)

**Proof:** It follows from Equation (36) that \( \hat{V}(\delta(t), v(t)) \leq 0 \) for all \( t \geq 0 \), and hence,

\[
V(\delta(t), v(t)) \leq V(\delta(0), v(0)), \quad t \geq 0.
\] (38)

Now, using the fact that \( V(\delta(t), v(t)) \geq \|\delta(t)\|_2, t \geq 0 \), Equation (38) yields

\[
\|\delta(t)\|_2 \leq \sqrt{V(\delta(0), v(0))}, \quad t \geq 0.
\] (39)

Since \( \| \cdot \|_{\infty} \leq \| \cdot \|_2 \) and every vector norm \( \| \cdot : \mathbb{R}^n \to \mathbb{R} \) is uniformly continuous on \( \mathbb{R}^n \), Equation (39) yields

\[
\|\delta\|_{\infty} \leq \sqrt{V(\delta(0), v(0))}.
\] (40)

Now, Equation (37) is a direct consequence of Equation (40), since Equation (40) holds uniformly in \( T > 0 \), \( \delta(t) = x(t) - av(x(t))e_i, \) and \( av(x(t)) = \frac{1}{n} e_i e_i^T x_0, t \geq 0 \).
Remark 5.2: Note that it follows from Equations (25) and (37) that the worst-case transient performance bound for an agent $i$ is given by

$$
\|x_i\|_{L_\infty} \leq \frac{1}{n} \|e_n e_n^T x_0\|_2 + \sqrt{V \left( \left[ I_n - \frac{1}{n} e_n e_n^T \right] x_0, v(0) \right)}.
$$

(41)

6. Illustrative numerical examples

In this section, we consider a networked multi-agent system represented by the undirected graph shown in Figure 1 for our first two examples and then consider three networked Boeing 747 airplanes on an undirected line graph for our third example. Our aim is to compare the performance of the standard consensus protocol given by Equation (10) with the proposed consensus protocols (13)–(15).

Example 6.1: For the undirected graph shown in Figure 1, let

$$
x_0 = \begin{bmatrix} 6, -5, 4, -3, 2, -1, 1, -2, 3, -4, 5, -6 \end{bmatrix}^T.
$$

(42)

Furthermore, let $k_1 = 7.5$ for the standard consensus protocol given by Equation (10) and let $k_1 = 2.5, k_2 = 0.001, \tau = 0.0001, \xi = 100, \theta = 1$ for the proposed consensus protocol given by Equations (13), (14), and (15). Figures 5 and 6 show the results for the standard and proposed consensus protocols, respectively. Note that even though both protocols achieve similar performance in terms of settling time, the control effort of the latter protocol is significantly less in magnitude as compared to the former protocol.

Example 6.2: For the undirected graph shown in Figure 1, let the (random) initial condition be given by

$$
x_0 = [-5.33, 4.67, 1.75, -0.15, 13.9, -7.83, -0.42, 9.02, 2.49, 0.21, -3.67, 16.15]^T.
$$

(43)

Furthermore, let $k_1 = 9$ for the standard consensus protocol given by Equation (10) and let $k_1 = 4.5, k_2 = 0.003, \tau = 0.0001, \xi = 100, \theta = 1$ for the proposed consensus protocol given by Equations (13), (14), and (15). Figures 7 and 8 show the results for the standard and proposed consensus protocols, respectively. Once again, both protocols achieve similar performance in terms of settling time; however, the control effort of the proposed protocol is less in magnitude as compared to the standard protocol.

Example 6.3: In this example, we use the proposed architecture for pitch rate consensus of commercial airplanes. Specifically, consider the multi-agent system representing the controlled longitudinal motion of three Boeing 747 airplanes (Bryson, 1993) linearised at an altitude of 40 kft and a velocity of 774 ft/sec given by

$$
\dot{\zeta}_i(t) = A\zeta_i(t) + B\delta_i(t), \quad \zeta_i(0) = \zeta_{i_0}, \quad i = 1, 2, 3, \quad t \geq 0,
$$

(44)

Figure 5. Agent and control responses for the standard consensus protocol given by Equation (10) with $k_1 = 7.5$ (Example 6.1).
where $\zeta_i(t) = [v_{x_i}(t), v_{z_i}(t), q_i(t), \theta_{e_i}(t)]^T \in \mathbb{R}^4$, $t \geq 0$, is state vector of agent $i$, $i = 1, 2, 3$, with $v_{x_i}(t), t \geq 0$, representing the $x$-body-axis component of the velocity of the airplane centre of mass with respect to the reference axes (in ft/sec), $v_{z_i}(t), t \geq 0$, representing the $z$-body-axis component of the velocity of the airplane centre of mass with respect to the reference axes (in ft/sec), $q_i(t), t \geq 0$, representing the $y$-body-axis component of the angular velocity of the airplane (pitch rate) with respect to the reference axes (in crad/sec), $\theta_{e_i}(t), t \geq 0$, representing the pitch Euler angle of the airplane body axes with respect to the reference axes (in crad), $\delta_i(t), t \geq 0$, representing the elevator control input.

Figure 6. Agent and control responses for the proposed consensus protocol with resetting given by Equations (13), (14), and (15) with $k_1 = 2.5, k_2 = 0.001, \tau = 0.0001, \xi = 100$, and $\theta = 1$ (Example 6.1).

Figure 7. Agent and control responses for the standard consensus protocol given by Equation (10) with $k_1 = 9$ (Example 6.2).
Figure 8. Agent and control responses for the proposed consensus protocol with resetting given by Equations (13), (14), and (15) with $k_1 = 4.5$, $k_2 = 0.003$, $\tau = 0.0001$, $\xi = 100$, and $\theta = 1$ (Example 6.2).

Figure 9. Agent guidance state ($x_i(t), t \geq 0$), guidance input ($u_i(t), t \geq 0$), pitch rate ($q_i(t), t \geq 0$), and elevator control ($\delta_i(t), t \geq 0$) responses for the standard consensus protocol given by Equation (10) with $k_1 = 1$ (Example 6.3).
We propose a two-level control hierarchy composed of a lower level controller for command following and a higher level controller for pitch rate consensus of the three airplanes given by Equation (44). To address lower level controller design, let $x_i(t)$, $s_i(t)$, $i = 1, 2, 3$, $t \geq 0$, be a command generated by Equation (9) (i.e., the guidance command) and let $s_i(t)$, $i = 1, 2, 3$, $t \geq 0$, denote the integrator state satisfying

$$
\dot{s}_i(t) = E\zeta_i(t) - x_i(t), \quad s_i(0) = s_{i0}, \quad i = 1, 2, 3, \quad t \geq 0,
$$

(46)

where $E = [0, 0, 1, 0]$. Now, defining the augmented state $\bar{\zeta}(t) \triangleq [\zeta^T(t), s_i(t)]^T$, Equations (44) and (46) give

$$
\dot{\bar{\zeta}}_i(t) = \bar{A}\bar{\zeta}_i(t) + \bar{B}_1\delta_i(t) + \bar{B}_2x_i(t), \quad \bar{\zeta}_i(0) = \tilde{\zeta}_{i0},
$$

where

$$
\bar{A} \triangleq \begin{bmatrix} A & 0 \\ E & 0 \end{bmatrix}, \quad \bar{B}_1 \triangleq \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \bar{B}_2 \triangleq \begin{bmatrix} 0 \\ -I \end{bmatrix}. \quad (47)
$$

Furthermore, let the elevator control input be given by

$$
\delta(t) = -K \bar{\zeta}(t), \quad K = [-0.0157, 0.0831, -4.7557, -0.1400, -9.8603], \quad t \geq 0,
$$

(49)

which is designed based on an optimal linear-quadratic regulator.

For the higher level controller design, we first use the standard consensus protocol given by Equation (10) and then use the proposed consensus protocol given by Equations (13), (14), and (15) to generate $x_i(t)$, $t \geq 0$, which has a direct effect on the lower level controller design to achieve pitch rate consensus. Figures 9 and 10 present the results

![Figure 10](image-url)

Figure 10. Agent guidance state ($x_i(t)$, $t \geq 0$), guidance input ($u_i(t)$, $t \geq 0$), pitch rate ($q_i(t)$, $t \geq 0$), and elevator control ($\delta_i(t)$, $t \geq 0$) responses for the proposed consensus protocol with resetting given by Equations (13), (14), and (15) with $k_1 = 0.5$, $k_2 = 0.00025$, $\tau = 0.001$, $\xi = 250$, and $\theta = 1$ (Example 6.3).
for all initial conditions set to zero and $x_1(0) = 8$, $x_2(0) = 4$, and $x_3(0) = 2$. It can be seen from the figures that the aircrafts reach a pitch rate consensus faster with less control effort ($u(t)$, $t \geq 0$) with the proposed consensus protocol shown in Figure 10 as compared with the standard consensus protocol shown in Figure 9.

7. Conclusion
This paper unifies the notions of hybrid resetting control and quasi-resetting control using a continuous controller architecture for multi-agent systems. In particular, we develop a consensus control protocol for networked multi-agent systems with a uniformly continuous quasi-resetting architecture that is free of well-posedness assumptions and time regularisation that is typically imposed by hybrid resetting controllers. Utilising Lyapunov–Krasovskii theory, these results are then used to show stability and asymptotic convergence of the multi-agent networked system while preserving the centroid of the network. Future extensions will focus on disturbance rejection and robustness properties of the proposed consensus protocols, as well as asynchronism, system time-delays, and switching network topologies for addressing possible information asynchrony between agents, message transmission and processing delays, and communication link failures and communication dropouts. In addition, we will explore extensions of the proposed framework to develop finite-time consensus protocols for addressing finite-time coordination, rather than merely asymptotic agreement.

Acknowledgements
The authors wish to thank Professor Magnus Egerstedt of the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, for several constructive suggestions.

Funding
This work was supported in part by the Air Force of Scientific Research under [grant number FA9550-12-1-0192].

Notes
1. Note that to implement Equation (6) exactly we need $\Delta t \to 0$, which implies $\epsilon \to 0$.
2. In this paper, we assume that the network is static, and hence, agent evolution will not cause edges to appear or disappear in the network. Thus, the proposed architecture does not address information link failures and communication dropouts.
3. The impulsive resetting controller given by Equations (20) and (21) can be viewed as a multi-agent system version of the Clegg integrator (Clegg, 1958).

References