

Chapter 1

Networked Hybrid Dynamical Systems: Models, Specifications, and Tools

Ricardo G. Sanfelice *

Abstract

Models, specifications, and tools for networked hybrid dynamical systems are presented. The proposed modeling framework allows the agent, the network, and the algorithms to have hybrid dynamics. Notions that properly capture key specifications for networked systems, namely, formation, synchronization, safety, and security, are provided. Tools for analysis of the closed-loop hybrid system and for the design of distributed hybrid algorithms are presented. Applications of the methods to estimation, consensus, and synchronization over complex networks are presented throughout the paper.

1.1 Introduction

The objective of this paper is to present mathematical models, specifications, notions, and tools for the design of algorithms for networked hybrid dynamical systems. A network of such systems is defined as multiple agents running algorithms that are allowed to share information over a network so as to fulfill a given design specification. The mathematical models of the agents, the algorithms, and the network are all given in terms of hybrid inclusions. In the autonomous case, a hybrid inclusion is given by

$$\begin{aligned} \dot{x} &\in F(x) & x &\in C \\ x^+ &\in G(x) & x &\in D \end{aligned} \tag{1.1}$$

where x is the state. This model allows the state to change continuously according to the constrained differential inclusion in (1.1) during flows and,

* R. G. Sanfelice is with the Department of Computer Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: ricardo@ucsc.edu).

at jumps, change discretely according to the constrained difference inclusion in (1.1). With such a general model, the agents may have states that evolve continuously and discretely, the models of the algorithms can have logic statements and conditions under which their response changes, and the models of the network may capture the conditions triggering communication events for agents to exchange information over the network. Due to the combination of heterogeneous continuous and discrete dynamics, the analysis of the resulting system as well as the design of algorithms and system parameters to satisfy a particular design specification cannot be carried out with tools for purely continuous-time or discrete-time systems.

Several unique features of networked hybrid dynamical systems makes their analysis and design challenging. These features include the unavoidable effect of the network, which, in most cases, does not allow continuous exchange of information, perturbations, and the inherently hybrid dynamics of the agents. More precisely:

1. *Distributed agents with hybrid dynamics*: the intervals of time over which the state of the agents changes continuously may be different among agents. The time instants at which the state of the agents change discretely may also not be the same. In fact, the assumption that all of the agents flow and jump at the same time might be too restrictive.
2. *Asynchronous communication events at unknown times*: the time instances at which agents exchange information may not be synchronized, meaning that each agent may receive information at different time instances. Furthermore, as in the previous item, the amount of ordinary time elapsed between communication events for each agent might be different; for instance, an agent can receive information at a much faster rate than others. In addition, the exact times at which information is exchanged may not be known a priori.
3. *Lack of full information at the same time*: The information about the states of the neighboring agents may not be available at the same time. In fact, most realistic models of networks would not provide information continuously, but rather, at isolated time instances. To met certain design specifications, such a constraint may require algorithms that can cope with limited information, both in terms of its value and the time information is received.
4. *Perturbations in the dynamics, parameters, and measurements*: the lack of knowledge of the actual models of each component of a network of hybrid systems would prevent one from compensating for their effect at the design stage. Designs that are robust to perturbations such as measurement noise, unmodeled dynamics, and delay are mandatory.

Section 1.2 of this paper pertains to modeling of networked hybrid dynamical systems. A mathematical model of each of the agents is introduced first. Each agent is modeled as a hybrid system, similar to (1.1). Such a model is general enough to allow for nonlinear, nonautonomous, set-valued,

and heterogeneous dynamics with solutions that evolve continuously and, at times, jump. Hybrid dynamical models capturing the mechanisms behind the networks connecting the agents are presented. The hybrid dynamics in these models allows to capture the discrete nature of communication events in digital networks. These models are also modular to permit their use in the definition of interconnections between the agents in the network, where its topology is defined by a graph. Finally, a general model of hybrid algorithms for the control of the agents is given in terms of hybrid inclusions as in (1.1) as well.

The interconnection between the agents, the networks, and the algorithms defines a closed-loop system that, after appropriate design, is to meet certain given specifications. With such a model at hand, Section 1.3 introduces specifications that are of typical interest in networked systems problems. These specifications are given in terms of the dynamical properties of the resulting closed-loop system. The property of all agents converging to a desired relative configuration, typically referred to as *formation*, is introduced as the property that solutions converge (in the limit or in finite time) to the set of points defining the formation. Synchronization is defined as the property that all solutions (or some of its components) converge to each other, property that we define as *asymptotic synchronization*, potentially with stability, which we refer to as *stable synchronization*. In addition, specifications that capture safety and security are also presented.

With the specifications introduced in Section 1.3, notions and tools that can be used to satisfy the given specifications are introduced in Section 1.4. The notions include asymptotic stability, finite time convergence, forward invariance, and robustness. Due to space constraints, we provide pointers to the literature of hybrid dynamical systems where formal statements and further applications of these notions and tools can be found. These methods have been recently used to solve problems pertaining to certain classes of networked hybrid dynamical systems, specifically, to solve state estimation [1, 2], consensus [3], synchronization [4, 5, 6, 7], and security [8] problems over networks. Section 1.5 provides a summary of some of these applications.

1.2 Networked Hybrid Dynamical Systems

In this section, we introduce a general model of N networked hybrid systems. A graph defines the network structure, in particular, the nodes and the communication links between them. Each node in the graph corresponds to an agent with general hybrid dynamics. The exchange of information between the agents is also modeled as a hybrid system, in particular, to capture the events at which communication events occur. Each agent is controlled by an algorithm that may also be hybrid.

1.2.1 Agents

For each $i \in \mathcal{V} := \{1, 2, \dots, N\}$, the i -th agent is modeled as a hybrid system \mathcal{H}_i^a with data $(C_i^a, F_i^a, D_i^a, G_i^a, E_i^a, H_i^a)$ and given by the hybrid inclusion with inputs and outputs

$$\begin{aligned} \dot{z}_i &\in F_i^a(z_i, u_i) & (z_i, u_i) &\in C_i^a \\ z_i^+ &\in G_i^a(z_i, u_i) & (z_i, u_i) &\in D_i^a \\ y_i &\in H_i^a(z_i, u_i) & (z_i, u_i) &\in E_i^a \end{aligned} \quad (1.2)$$

where $z_i \in \mathbb{R}^{n_i^a}$ is the state, $u_i \in \mathbb{R}^{m_i^a}$ the input, and $y_i \in \mathbb{R}^{p_i^a}$ the output of the i -th agent. The set-valued map F_i^a is the flow map capturing the continuous dynamics and C_i^a defines the flow set on which flows are allowed. The set-valued map G_i^a defines the jump map and models the discrete behavior, and D_i^a defines the jump set, which is where jumps are allowed. The set E_i^a defines the output set. A solution² to \mathcal{H}_i^a is given by a pair (ϕ_i, u_i) parametrized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, where t denotes ordinary time and j denotes jump time. The domain $\text{dom}(\phi_i, u_i) \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a hybrid time domain if for every $(T, J) \in \text{dom}(\phi_i, u_i)$, the set $\text{dom}(\phi_i, u_i) \cap ([0, T] \times \{0, 1, \dots, J\})$ can be written as $\cup_{j=0}^J (I_j \times \{j\})$, where $I_j := [t_j, t_{j+1}]$ for a time sequence $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_J \leq t_{J+1}$. The t_j 's with $j > 0$ define the time instants when the state of the hybrid system jumps and j counts the number of jumps. The set $\mathcal{S}_{\mathcal{H}_i^a}$ contains all maximal solutions to \mathcal{H}_i^a , and the set $\mathcal{S}_{\mathcal{H}_i^a}(\xi)$ contains all maximal solutions to \mathcal{H}_i^a with initial condition ξ .

Example 1. A widely studied problem in the literature of multi-agent systems is the problem of controlling the state of point-mass systems over a network to reach consensus. In such a case, the dynamics of the agents are simply $\dot{z}_i = u_i$ for each $i \in \mathcal{V}$, where $z_i, u_i \in \mathbb{R}^{n_i^a}$ for some $n_i^a = m_i^a$. Certainly, such dynamics can be modeled as shown in (1.2) by choosing $F_i^a(z_i, u_i) := u_i$, $G_i^a(z_i, u_i)$ arbitrary, $C_i^a = \mathbb{R}^{n_i^a} \times \mathbb{R}^{n_i^a}$, and D_i^a empty. The model in (1.2) also allows to include constraints in the state and the input of each agent. For instance, if the input of the agent is constrained to $|u_i| \leq \bar{u}$ for some $\bar{u} > 0$ then the flow set can be defined as $C_i^a = \mathbb{R}^{n_i^a} \times \{u_i \in \mathbb{R}^{m_i^a} : |u_i| \leq \bar{u}\}$. More interestingly, the model in (1.2) permits capturing agents with point-mass hybrid dynamics, such as

$$\dot{z}_i = u_{i,1} =: F_i^a(z_i, u_i)$$

during flows and

² A solution to \mathcal{H}_i^a is called maximal if it cannot be extended, i.e., it is not a (proper) truncated version of another solution. It is called complete if its domain is unbounded. A solution is Zeno if it is complete and its domain is bounded in the t direction. A solution is precompact if it is complete and bounded.

$$z_i^+ = u_{i,2} =: G_i^a(z_i, u_i)$$

at jumps, where $u_i = (u_{i,1}, u_{i,2})$. In such a model, the conditions on the state and the input imposed by the flow and jump sets would determine when the input $u_{i,1}$ affecting the flows is active, and when the input $u_{i,2}$ assigning the state after jumps is active. ■

Example 2. Synchronization of the state of nonlinear continuous-time systems of the form $\dot{z}_i = f_i(z_i, u_i)$ emerges in many problems in science and engineering. Such an agent model is captured by defining $F_i^a(z_i, u_i) := f_i(z_i, u_i)$, $G_i^a(z_i, u_i)$ arbitrary, $C_i^a = \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$, and D_i^a empty. More interestingly, the model in (1.2) allows for jumps in the state that can emerge due to hybrid dynamics in the agents themselves. The mathematical models of impulse-coupled oscillators used in the literature to capture the dynamics of populations of fireflies and neurons exhibit such dynamics; see, e.g., [9]. For instance, one such a model consists of a scalar state z_i of each oscillator taking values in the compact set $[0, T]$, where $T > 0$ is a parameter, and that, during flows, increases monotonically towards T . During this regime, the change of z_i is governed by the autonomous system $\dot{z}_i = f_i(z_i)$ and the state z_i is constrained to $[0, T]$. Upon reaching a threshold T , the state z_i self-resets to zero. Furthermore, when agents that are neighbors to the i -th agent self-reset their states to zero, they trigger a reset of the state z_i to a value that may depend on the state of the i -th agent and of its neighbors. Letting u_i be the input to the i -th agent, which is to be assigned to a function of the state of the neighbors so as to externally reset z_i as just described, the change of z_i at self-triggered jumps is $z_i^+ = 0$ and at externally-triggered jumps as $z_i^+ = g_i(z_i, u_i)$. An agent model as in (1.2) is given by

$$\begin{aligned} \dot{z}_i &= f_i(z_i) =: F_i^a(z_i, u_i) & (z_i, u_i) &\in [0, T] \times \mathbb{R}^{m_i} =: C_i^a, \\ z_i^+ &\in G_i^a(z_i, u_i) := \begin{cases} 0 & \text{if } z_i = T, u_i \notin D_i^e \\ g_i(z_i, u_i) & \text{if } z_i \in [0, T), u_i \in D_i^e \\ \{0, g_i(z_i, u_i)\} & \text{if } z_i = T, u_i \in D_i^e \end{cases} \\ & & (z_i, u_i) &\in (\{T\} \times \mathbb{R}^{m_i}) \cup ([0, T] \times D_i^e) =: D_i^a, \\ y_i &= z_i =: H_i^a(z_i, u_i) & (z_i, u_i) &\in [0, T] \times \mathbb{R}^{m_i} =: E_i^a \end{aligned}$$

In this model, the events are triggered when $z_i = T$ or u_i is equal to a value, or more generally, belong to an appropriately defined set describing the conditions that externally reset z_i . The latter set is denoted as D_i^e in the model above. Note that when both reset conditions occur simultaneously, the jump map of the agent is set valued, meaning that either one of the two possible resets is possible. ■

1.2.2 Networks

A directed graph (digraph) is defined as $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{G})$. The set of nodes of the digraph are indexed by the elements of \mathcal{V} and the edges are pairs in the set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Each edge directly links two different nodes, i.e., an edge from i to k , denoted by (i, k) , implies that agent i can send information to agent k . The adjacency matrix of the digraph Γ is denoted by $\mathcal{G} \in \mathbb{R}^{N \times N}$, whose entries g_{ik} take values on $\{0, 1\}$ according to the connectivity map: $g_{ik} = 1$ if $(i, k) \in \mathcal{E}$, and $g_{ik} = 0$ otherwise. The set of indices corresponding to the neighbors that can send information to the i -th agent is denoted by $\mathcal{N}(i) := \{k \in \mathcal{V} : (k, i) \in \mathcal{E}\}$. The in-degree and out-degree of agent i are defined by $d_i^{\text{in}} = \sum_{k=1}^N g_{ki}$ and $d_i^{\text{out}} = \sum_{k=1}^N g_{ik}$. The in-degree matrix \mathcal{D} is the diagonal matrix with entries $D_{ii} = d_i^{\text{in}}$ for all $i \in \mathcal{V}$. The Laplacian matrix of the digraph Γ , denoted by $\mathcal{L} \in \mathbb{R}^{N \times N}$, is defined as $\mathcal{L} = \mathcal{D} - \mathcal{G}$. A digraph is said to be

- *weight balanced* if, at each node $i \in \mathcal{V}$, the out-degree and in-degree are equal; i.e., for each $i \in \mathcal{V}$, $d_i^{\text{out}} = d_i^{\text{in}}$;
- *completely connected* if every pair of distinct vertices is connected by a unique edge; that is $g_{ik} = 1$ for each $i, k \in \mathcal{V}$, $i \neq k$;
- *strongly connected* if and only if any two different nodes of the digraph can be connected via a path that traverses the directed edges of the digraph.

In most applications involving networks, the transfer of information between neighboring agents is driven by events. The events triggering communication between neighboring agents may depend on the state, the input, output information, or on a local quantity. The following general hybrid system model, denoted $\mathcal{H}_{ik}^{\text{net}}$, is used to trigger such events for each $(i, k) \in \mathcal{E}$:

$$\begin{aligned} \dot{\mu}_{ik} &\in F_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) & (\mu_{ik}, \omega_{ik}) &\in C_{ik}^{\text{net}} \\ \mu_{ik}^+ &\in G_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) & (\mu_{ik}, \omega_{ik}) &\in D_{ik}^{\text{net}} \\ \chi_{ik} &\in H_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) & (\mu_{ik}, \omega_{ik}) &\in E_{ik}^{\text{net}} \end{aligned} \quad (1.3)$$

where $\mu_{ik} \in \mathbb{R}^{n_{ik}^{\text{net}}}$ is a state variable associated to the communication of information from agent i to agent k , $\omega_{ik} \in \mathbb{R}^{m_{ik}^{\text{net}}}$ is its input, which might be assigned to information that agent i has to transmit to agent k as well as state variables in agent i that determine whether μ_{ik} should evolve continuously or discretely, and $\chi_{ik} \in \mathbb{R}^{p_{ik}^{\text{net}}}$ is its output, which includes the information that is transmitted from agent i to agent k . The hybrid model $\mathcal{H}_{ik}^{\text{net}}$ is general enough to capture most communication mechanisms or protocols in the literature. The following sample-and-hold mechanism defines perhaps the simplest version of a model to trigger communication of information from agent i to agent k .

Example 3 (periodic communication events with memory). The simplest event-driven communication protocol is perhaps one that collects information and

transmits it periodically. Let $T > 0$ denote the period for the events. A model that, after the first event, updates the information provided by the network after every T seconds have elapsed can be modeled as (1.3) for each $(i, k) \in \mathcal{E}$. Let τ_{ik} denote a timer state that triggers the communication events and let ℓ_{ik} be a memory state that stores the information at those events. Then, defining the state of (1.3) as $\mu_{ik} = (\tau_{ik}, \ell_{ik})$, the following model captures the network described above:

$$\begin{aligned} \dot{\mu}_{ik} &= \begin{bmatrix} \dot{\tau}_{ik} \\ \dot{\ell}_{ik} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} && \text{when } \tau_{ik} \in [0, T] \\ \mu_{ik}^+ &= \begin{bmatrix} \tau_{ik}^+ \\ \ell_{ik}^+ \end{bmatrix} = \begin{bmatrix} 0 \\ \omega_{ik} \end{bmatrix} && \text{when } \tau_{ik} = T \end{aligned} \quad (1.4)$$

where ω_{ik} is the input to the network, which has the information to communicate, and the output is $\chi_{ik} = \ell_{ik}$. Then, the data of $\mathcal{H}_{ik}^{\text{net}}$ is given by

$$\begin{aligned} F_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} \in [0, T]\} \\ G_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= \begin{bmatrix} 0 \\ \omega_{ik} \end{bmatrix} \\ D_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} = T\} \\ H_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= \ell_{ik} \\ E_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} \in [0, T]\} \end{aligned}$$

A network model in which collection and transmission of information do not occur simultaneously can be obtained by adding a timer and a memory state to the model above. In such a model, one of the timers, denoted as $\tau_{ik,1}$, triggers the events every T_1 seconds, at which events the input ω_{ik} is stored in a memory state, denoted as $\ell_{ik,1}$. The other timer, denoted as $\tau_{ik,2}$, triggers the events every T_2 seconds updating the memory state assigning the output, denoted $\ell_{ik,2}$, to the recorded value of ω_{ik} in $\ell_{ik,1}$. A model as in (1.3) capturing such mechanism has state $\mu_{ik} = (\tau_{ik,1}, \ell_{ik,1}, \tau_{ik,2}, \ell_{ik,2})$ and data

$$\begin{aligned}
F_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= (1, 0, 1, 0) \\
C_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik,1} \in [0, T_1], \tau_{ik,2} \in [0, T_2]\} \\
G_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= \begin{cases} (0, \omega_{ik}, \tau_{ik,2}, \ell_{ik,2}) & \text{if } \tau_{ik,1} = T_1, \tau_{ik,2} \in [0, T_2), \\ (\tau_{ik,1}, \ell_{ik,1}, 0, \ell_{ik,1}) & \text{if } \tau_{ik,1} \in [0, T_1), \tau_{ik,2} = T_2, \\ \{(0, \omega_{ik}, \tau_{ik,2}, \ell_{ik,2}), (\tau_{ik,1}, \ell_{ik,1}, 0, \ell_{ik,1})\} & \text{if } \tau_{ik,1} = T_1, \tau_{ik,2} = T_2, \end{cases} \\
D_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik,1} = T_1\} \cup \{(\mu_{ik}, \omega_{ik}) : \tau_{ik,2} = T_2\} \\
H_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= \ell_{ik} \\
E_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik,1} \in [0, T_1], \tau_{ik,2} \in [0, T_2]\}
\end{aligned}$$

The jump set D_{ik}^{net} of this hybrid system captures the two possible events, which are when $\tau_{ik,1} = T_1$ or when $\tau_{ik,2} = T_2$, and the jump map G_{ik}^{net} resets the state variables according to which event has occurred. Note that when both events happen simultaneously, the jump map is set valued. In general, the parameters T_1 and T_2 in the models above may depend on each agent, in which case they will be denoted as T_1^i and T_2^i . ■

While the models in Example 3 capture the key property that information transmitted over networks is typically only available at isolated time instances, they make the assumption that transmissions occur periodically. The following model relaxes that assumption by allowing consecutive communication events to occur within a window of finite length. For simplicity, this extension is carried out for the model in (1.4) and without a memory state. An extension for the case with memory states and two timers follows similarly.

Example 4 (aperiodic communication events). The first model in Example 3 guarantees that every solution has a hybrid time domain defined by a sequence $t_0 = 0 \leq t_1 < t_2 < t_3 < \dots$ satisfying

$$t_{j+1} - t_j = T$$

for all $j > 0$ such that (t, j) is in the domain of the solution. When the time in between consecutive events is not constant, but rather known to occur no later than T_2 seconds and no sooner than T_1 seconds after every event, the sequence of times $\{t_j\}$ would satisfy

$$t_{j+1} - t_j \in [T_1, T_2] \tag{1.5}$$

for all $j > 0$ such that (t, j) is in the domain of the solution. The parameters T_1 and T_2 are such that $T_2 \geq T_1 > 0$. In principle, the event times t_j can be thought of being determined by a random variable taking values in the interval $[T_1, T_2]$. The following model generates solutions satisfying (1.5) by exploiting nondeterministic behavior due to an overlap between the flow and jump sets:

$$\dot{\tau}_{ik} = 1 \quad \tau_{ik} \in [0, T_2] \quad (1.6a)$$

$$\tau_{ik}^+ = 0 \quad \tau_{ik} \in [T_1, T_2] \quad (1.6b)$$

In fact, whenever the timer state τ_{ik} is in $[T_1, T_2]$, both flows and jumps are possible, meaning that there exist solutions that jump or that flow when τ_{ik} is equal to any point in that set. A model as in (1.3) capturing such mechanism has state $\mu_{ik} = \tau_{ik}$, and flow and jump maps/sets given by

$$\begin{aligned} F_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= 1 \\ C_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} \in [0, T_2]\} \\ G_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= 0 \\ D_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} \in [T_1, T_2]\} \end{aligned}$$

An alternative model that generates solutions satisfying (1.5) but, instead, through a set-valued jump map, is given by

$$\dot{\tau}_{ik} = -1 \quad \tau_{ik} \in [0, T_2] \quad (1.7a)$$

$$\tau_{ik}^+ \in [T_1, T_2] \quad \tau_{ik} = 0 \quad (1.7b)$$

In this model, the communication events are triggered by a timer τ_{ik} that decreases and upon reaching zero, it is reset to a point in $[T_1, T_2]$. The hybrid system (1.7) can be captured by (1.3) by choosing the state as $\mu_{ik} = \tau_{ik}$, and flow and jump maps/sets given by

$$\begin{aligned} F_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= -1 \\ C_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} \in [0, T_2]\} \\ G_{ik}^{\text{net}}(\mu_{ik}, \omega_{ik}) &:= [T_1, T_2] \\ D_{ik}^{\text{net}} &:= \{(\mu_{ik}, \omega_{ik}) : \tau_{ik} = 0\} \end{aligned}$$

■

Network mechanisms and protocols that employ timers, memory states, and logic can be fit in the network model (1.3), for instance, TCP/IP [10], wireless Ethernet, and bluetooth protocols [11] can be modeled with such a hybrid model.

1.2.3 Algorithms

For each $i \in \mathcal{V}$, the algorithm associated to the i -th agent is modeled as a hybrid system \mathcal{H}_i^K with data $(C_i^K, F_i^K, D_i^K, G_i^K, E_i^K, H_i^K)$ and given by the hybrid inclusion with inputs and outputs

$$\begin{aligned}
\dot{\eta}_i &\in F_i^K(\eta_i, v_i) & (\eta_i, v_i) &\in C_i^K \\
\eta_i^+ &\in G_i^K(\eta_i, v_i) & (\eta_i, v_i) &\in D_i^K \\
\zeta_i &\in H_i^K(\eta_i, v_i) & (\eta_i, v_i) &\in E_i^K
\end{aligned} \tag{1.8}$$

where $\eta_i \in \mathbb{R}^{n_i^K}$ is the state, $v_i \in \mathbb{R}^{m_i^K}$ the input, and $\zeta_i \in \mathbb{R}^{p_i^K}$ the output of the i -th agent. As for \mathcal{H}_i^a in (1.2), the set-valued map F_i^K is the flow map capturing the continuous dynamics and C_i^K defines the flow set on which flows are allowed. The set-valued map G_i^K defines the jump map and models the discrete behavior, and D_i^K defines the jump set which is where jumps are allowed. The set E_i^K defines the output set. A solution to the algorithm \mathcal{H}_i^K can also be defined, as done for \mathcal{H}_i^a .

Example 5. Algorithms that, at isolated time instants, measure the received data and compute a feedback control law that is to be applied to the agent can be modeled as the algorithm in (1.8). An algorithm with sampling events triggered when one of its inputs reaches a particular value, at which event uses the information in another of its inputs to compute the control law and stores it in a memory state that is to be applied to the agent is given as follows. Let $v_i = (v_{i,1}, v_{i,2})$ be the input to the algorithm, where $v_{i,1}$ is the input triggering the events and $v_{i,2}$ is the input with the information needed to compute the control law. Suppose that $v_{i,1}$ reaching zero triggers the computation events. Let ℓ_i be a state variable that, at the events, stores the value of the feedback control law, which is given by the function κ , and in between events remains constant. The discrete dynamics of the algorithm are

$$\ell_i^+ = \kappa(v_{i,2})$$

which are active when $v_{i,1} = 0$. The continuous dynamics of the algorithm are simply

$$\dot{\ell}_i = 0$$

which, in principle,³ are active when $v_{i,1} > 0$. In this way, the state ℓ_i operates as a memory state. This algorithm is given by \mathcal{H}_i^K as in (1.8) with state $\eta_i = \ell_i$, input $v_i = (v_{i,1}, v_{i,2})$, and data given by

$$\begin{aligned}
F_i^K(\eta_i, v_i) &= 0 \\
C_i^K &= \{(\eta_i, v_i) : v_{i,1} > 0\} \\
G_i^K(\eta_i, v_i) &= \kappa(v_{i,2}) \\
D_i^K &= \{(\eta_i, v_i) : v_{i,1} = 0\} \\
H_i^K(\eta_i, v_i) &= \ell_i
\end{aligned}$$

and E_i^K the entire state and input space of \mathcal{H}_i^K .

³ In Section 1.5.1, we present a model for which the continuous dynamics are active when $v_{i,1} \geq 0$ due to such input being connected to a strictly decreasing timer, in which case, flows with $v_{i,1}$ identically zero are not possible. In such a case, the set C_i^K is closed.

The model for \mathcal{H}_i^K also allows to capture the dynamics behind an algorithm that does not trigger computations synchronously with the arrival of information. The computation events in such algorithm could be triggered by an internal state, at which events the last piece of information received is used to compute the feedback law. Such a mechanism can be modeled using a memory state that stores the information received, a memory state that stores the computed feedback law, and a state that triggers the events. Denote these state variables as $\ell_{i,1}$, $\ell_{i,2}$, and τ_i , respectively, which define the state of the algorithm $\eta_i = (\ell_{i,1}, \ell_{i,2}, \tau_i)$. As in the model with a single memory state given above, the memory state $\ell_{i,1}$ stores the information in $v_{i,2}$ when the input $v_{i,1}$ reaches zero. Let $\tilde{\gamma}$ be a function that, when zero, triggers the computation of the feedback control law, which is denoted as κ . Then, the discrete dynamics of τ_i are active when

$$\tilde{\gamma}(\eta_i) = 0$$

At each jump, the discrete dynamics update τ_i according to

$$\tau_i^+ = \varrho_i^d(\eta_i)$$

where ϱ_i^d is a function to be defined, and $\ell_{i,2}$ according to

$$\ell_{i,2}^+ = \kappa(\ell_{i,1})$$

We assume that flows of τ_i are active when

$$\tilde{\gamma}(\eta_i) \geq 0$$

and that are governed by

$$\dot{\tau}_i = \varrho_i^c(\eta_i)$$

The function ϱ_i^c is assumed to not allow flows that remain in $\tilde{\gamma}(\eta_i) = 0$. The memory state $\ell_{i,2}$ remains constant during flows. This algorithm is given by \mathcal{H}_i^K as in (1.8) with state $\eta_i = (\ell_{i,1}, \ell_{i,2}, \tau_i)$, input $v_i = (v_{i,1}, v_{i,2})$, and data given by

$$\begin{aligned} F_i^K(\eta_i, v_i) &= (0, 0, \varrho_i^c(\eta_i)) \\ C_i^K &= \{(\eta_i, v_i) : v_{i,1} > 0, \tilde{\gamma}(\eta_i) \geq 0\} \\ G_i^K(\eta_i, v_i) &= \begin{cases} (v_{i,1}, \ell_{i,2}, \tau_i) & \text{if } v_{i,2} = 0, \tilde{\gamma}(\eta_i) > 0 \\ (\ell_{i,1}, \kappa(\ell_{i,1}), \varrho_i^d(\eta_i)) & \text{if } v_{i,1} > 0, \tilde{\gamma}(\eta_i) = 0 \\ \{(v_{i,2}, \ell_{i,2}, \tau_i), (\ell_{i,1}, \kappa(\ell_{i,1}), \varrho_i^d(\eta_i))\} & \text{if } v_{i,1} = 0, \tilde{\gamma}(\eta_i) = 0 \end{cases} \\ D_i^K &= \{(\eta_i, v_i) : v_{i,1} = 0\} \cup \{(\eta_i, v_i) : \tilde{\gamma}(\eta_i) = 0\} \\ H_i^K(\eta_i, v_i) &= \ell_{i,2} \end{aligned}$$

and E_i^K the entire state and input space of \mathcal{H}_i^K . ■

The model in \mathcal{H}_i^K is general enough to allow for multi-mode, event-triggered, and predictive-based algorithms.

1.2.4 Closed-loop System

Given a digraph Γ , the interconnection between the agents, algorithms, and network models results in a hybrid system. Assuming that, for each $i \in \mathcal{V}$, the input u_i of the i -th agent is assigned to the output ζ_i of the i -th algorithm, that the input v_i of the i -th algorithm is assigned to a function of y_i and of the output of the networks connected to it, namely $\{\chi_{ki}\}_{k \in \mathcal{N}(i)}$, and that, for each $k \in \mathcal{N}(i)$, the input ω_{ik} is assigned to y_i , the interconnection between these hybrid systems lead to an autonomous hybrid system \mathcal{H} of the form

$$\begin{aligned} \dot{x} &\in F(x) & x &\in C \\ x^+ &\in G(x) & x &\in D \end{aligned} \tag{1.9}$$

where

$$x = (x_1, x_2, \dots, x_N) \in \mathbb{R}^n$$

is the state with $n = \sum_{i \in \mathcal{V}} (n_i^a + n_i^K + d_i^{\text{in}} n_{ki}^{\text{net}})$, where x_i collects the states components of the agent, algorithm, and networks associated to the i -th agent. The data (C, F, D, G) is constructed using the data of the individual systems. In Section 1.5 we provide numerous examples of such construction.

1.3 Design Specifications

In this section, we formulate specific properties of interest in the design of networked systems. The network-specific properties introduced include the situation when the states of the individual systems reach a particular set that depends on the local variables, which we call *formation*, that the states of all systems converge to each other, which is referred to as *synchronization*, that the entire interconnected hybrid system is safe, called *safety*, and that exogenous signals injected at specific agents are detectable, which we refer to as *security*. These properties are given in terms of the variables and inputs of the individual agents.

1.3.1 Formation

A property that is of interest in network system problems is when, for each $i \in \mathcal{V}$, the state x_i converges to a particular relative configuration. For the

closed-loop system \mathcal{H} , the set of interest is given as

$$\mathcal{A} := \bigcap_{i \in \mathcal{V}} \mathcal{A}_i \quad (1.10)$$

where

$$\mathcal{A}_i := \{x \in \mathbb{R}^n : \rho_i(x) = 0\}$$

and, for each $i \in \{1, 2, \dots, N\}$, the function ρ_i defines the relative formation between the agent i and the other agents. Convergence of solutions to this set can be interpreted as the network reaching a formation, in particular, when components of x_i are related to physical quantities, such as position or angles. To formulate this property, denote the distance from x to \mathcal{A} as $|x|_{\mathcal{A}}$, namely,

$$|x|_{\mathcal{A}} = \inf_{x' \in \mathcal{A}} |x - x'|$$

Then the goal is to design an algorithm such that every maximal solution ϕ to \mathcal{H} converges to \mathcal{A} in finite time or asymptotically, that is, in the limit as “hybrid” time gets large:

- For some $(t^*, j^*) \in \text{dom } \phi$

$$\lim_{(t,j) \in \text{dom } \phi, t+j \searrow t^*+j^*} |\phi(t, j)|_{\mathcal{A}} = 0$$

- If ϕ is complete, then

$$\lim_{(t,j) \in \text{dom } \phi, t+j \rightarrow \infty} |\phi(t, j)|_{\mathcal{A}} = 0$$

Note that while in some network systems problems converging to the set \mathcal{A} might be possible without exchanging information between agents, there are numerous problems where transmission of information between agents and algorithms is mandatory. One such a case for $N = 2$ is when the algorithm that controls agent \mathcal{H}_1^a is \mathcal{H}_2^K , and the algorithm that controls agent \mathcal{H}_2^a is \mathcal{H}_1^K .

Certainly, the construction of the set \mathcal{A} in (1.10) covers the situation when state components of the algorithm for each agent are to converge to a common point, say z^* . In such a case, the definition of the sets \mathcal{A}_i will include the condition $z_i = z^*$ for each $i \in \mathcal{V}$. It also covers the setting when the algorithm reconstructs the state z_i from measurements of y_i , namely, the algorithm includes an observer. In such a case, the set \mathcal{A}_i will include a condition of the form $z_i = \hat{z}_i$, where \hat{z}_i is the component of η_i that provides an estimate of z_i .

1.3.2 Synchronization

Another dynamical property of interest in many network systems problems is when particular components of the solutions to each agent converge to each other, rather than to a particular set or point. For the closed-loop system \mathcal{H} , this property is stated as follows. Let $x = (x_1, x_2, \dots, x_N)$ be partitioned as $x_i = (p_i, q_i)$. The closed-loop system \mathcal{H} is said to have

- *stable synchronization with respect to p* if for every $\varepsilon > 0$, there exists $\delta > 0$ such that every maximal solution $\phi = (\phi_1, \phi_2, \dots, \phi_N)$, where $\phi_i = (\phi_{i,p}, \phi_{i,q})$, to \mathcal{H} such that

$$|\phi_i(0,0) - \phi_k(0,0)| \leq \delta$$

for each $i, k \in \mathcal{V}$ implies

$$|\phi_{i,p}(t,j) - \phi_{k,p}(t,j)| \leq \varepsilon$$

for all $i, k \in \mathcal{V}$ and $(t,j) \in \text{dom } \phi$.

- *globally attractive synchronization with respect to p* if every maximal solution is complete, and for each $i, k \in \mathcal{V}$

$$\lim_{\substack{(t,j) \in \text{dom } \phi \\ t+j \rightarrow \infty}} |\phi_{i,p}(t,j) - \phi_{k,p}(t,j)| = 0$$

- *global asymptotic synchronization with respect to p* if it has both stable synchronization and global attractive synchronization with respect to p .

In general, this is a partial state synchronization notion, but if $x_i = p_i$ for each $i \in \mathcal{V}$, then this notion can be considered to be a full-state synchronization notion. Note that stable synchronization with respect to p requires solutions ϕ_i for each $i \in \mathcal{V}$ to start close to each other, while only the components $\phi_{i,p}$, $i \in \mathcal{V}$ remain close to each other over their domain of definition. Similarly, global attractive synchronization with respect to p only requires that the Euclidean distance between each ϕ_i approaches zero, while the other components are left unconstrained. Also, note that boundedness of the solutions is not required.

1.3.3 Safety

Safety is a property of interest in the design of most algorithms for dynamical systems. Safety is typically characterized by conditions on the system variables, called *safety conditions*, that guarantee system operation within limits and away from undesired configurations. A system is said to be safe when its solutions are such that they remain within the set of points where the

safety conditions are satisfied. For each $i \in \mathcal{V}$, let K_i denote the set of points defining the safety conditions for the variables of the i -th agent and the set

$$K := K_1 \times K_2 \times \dots \times K_N$$

be the set that captures all safety conditions for the closed-loop system \mathcal{H} . Then, a particular safety goal is to design the algorithms and the networks such that every solution ϕ to \mathcal{H} with initial condition

$$\phi(0, 0) \in K$$

is such that

$$\phi(t, j) \in K \quad \forall (t, j) \in \text{dom } \phi$$

Note that this property enforces all solutions that start from K to remain in K , even if they are not complete. At times, one might be interested in the property that solutions starting from a potentially smaller set than K , stay in K . More precisely, let K_0 denote the set of allowed initial conditions. Then, such a safety property is as follows: design the algorithms and the networks such that every solution ϕ to \mathcal{H} with initial condition

$$\phi(0, 0) \in K_0$$

is such that

$$\phi(t, j) \in K \quad \forall (t, j) \in \text{dom } \phi$$

where, in most cases, the set K_0 would be strictly contained in the K .

1.3.4 Security

The general closed-loop system \mathcal{H} allows to model the dynamics of the physical components, such as sensors and actuators, the cyber components, which include digital devices and computing, as well as their interfaces. These interfaces can be exploited by adversaries to, for example, deny access or corrupt the information transmitted among agents. The characterization of which attacks are detectable and the design of algorithms to detect them are of great importance. Modeling the attacks as exogenous signals w_c and w_d affecting the continuous and discrete dynamics of \mathcal{H} , respectively, the closed loop under the effect of attacks is given by

$$\begin{aligned} \dot{x} &\in F(x + w_{c,1}) + w_{c,2} & x + w_{c,3} &\in C \\ x^+ &\in G(x + w_{d,1}) + w_{d,2} & x + w_{d,3} &\in D \end{aligned}$$

where $w_c = (w_{c,1}, w_{c,2}, w_{c,3})$ and $w_d = (w_{d,1}, w_{d,2}, w_{d,3})$. We refer to this closed-loop system as \mathcal{H}_w . In this context, the security problem consists of

detecting when the exogenous signal $w := (w_c, w_d)$ is nonzero. One way to accomplish that is to design a function that, when evaluated along solutions only, is nonzero if the attacker's input w is nonzero. For instance, one would be interested in designing a function r such that for every solution pair (ϕ, w) to \mathcal{H}_w

$$\exists(t, j) \in \text{dom}(\phi, w) : |w(t, j)| > 0 \quad \Rightarrow \quad |r(\phi(t, j))| > 0$$

Note that when the input w is nonzero over an interval, it might suffice to have a function r that becomes nonzero at some time over that interval, within some reasonable amount of time since the attack started.

1.4 Notions and Design Tools

In this section, we present dynamical properties that are suitable to certify the network-specific properties given in Section 1.3. These properties are stated for general hybrid systems given as \mathcal{H} in (1.9) and later, in Section 1.5, specialized to networked systems problems. The presentation of these properties and related results is informal, and pointers to the literature with formal statements are given.

1.4.1 Asymptotic Stability

Given a subset of the state space of a dynamical system, asymptotic stability captures the property that solutions starting close to the set stay close to it, and that solutions that are complete converge to it asymptotically. For a hybrid system \mathcal{H} as in (1.9) with state space \mathbb{R}^n , a closed set $\mathcal{A} \subset \mathbb{R}^n$ is said to be

- *stable* for \mathcal{H} if for each $\varepsilon > 0$ there exists $\delta > 0$ such that each solution ϕ to \mathcal{H} with initial condition such that

$$|\phi(0, 0)|_{\mathcal{A}} \leq \delta$$

satisfies

$$|\phi(t, j)|_{\mathcal{A}} \leq \varepsilon \quad \forall(t, j) \in \text{dom } \phi$$

- *globally asymptotically attractive* for \mathcal{H} if every maximal solution ϕ to \mathcal{H} is complete⁴ and satisfies

⁴ This attractivity notion enforces that every maximal solution to \mathcal{H} is complete, which is a property that is not for free. Sufficient conditions guaranteeing that maximal solutions are complete are given in [12, Proposition 2.10 and Proposition 6.10]. An attractivity notion

$$\lim_{(t,j) \in \text{dom } \phi, t+j \rightarrow \infty} |\phi(t, j)|_{\mathcal{A}} = 0$$

- *globally asymptotically stable* for \mathcal{H} if it is stable and globally asymptotically attractive.

Algorithms \mathcal{H}_i^K for \mathcal{H}_i^a that, under the effect of the networks $\mathcal{H}_{ik}^{\text{net}}$, guarantee asymptotic stability of the set \mathcal{A} can be designed using the Lyapunov stability analysis tools in [12, Chapter 3 and Chapter 7]. In particular, asymptotic stability is of interest to networked systems problems as it can be employed to guarantee formation and synchronization. In fact, the problem of guaranteeing that the network \mathcal{H} asymptotically reaches a formation can be solved by showing that the set \mathcal{A} in (1.10) is asymptotically attractive. The problem of designing algorithms \mathcal{H}_i^K that guarantee full state asymptotic synchronization of \mathcal{H} can be recast as the problem of asymptotically stabilizing the closed set

$$\mathcal{A} := \{x \in \mathbb{R}^n : x_1 = x_2 = \dots = x_N\}$$

Sufficient conditions for asymptotic stability in terms of Lyapunov functions can be found in [12, Chapters 3, 6, and 7]; see Section 1.5 for illustrations.

1.4.2 Finite time Convergence

At times, convergence to the set of points of interest in finite time is desired. For instance, in a network system one might be interested in assuring that the state of the individual systems converge to a particular formation, and after that, accomplish a different task. For a hybrid system \mathcal{H} on \mathbb{R}^n , given a closed set $\mathcal{A} \subset \mathbb{R}^n$, an open neighborhood \mathcal{N} of \mathcal{A} , and a function $\mathcal{T} : \mathcal{N} \rightarrow [0, \infty)$ called the settling-time function, the closed set \mathcal{A} is said to be

- *finite time attractive* for \mathcal{H} if each solution ϕ to \mathcal{H} with initial condition such that

$$\phi(0, 0) \in \mathcal{N}$$

satisfies

$$\sup_{(t,j) \in \text{dom } \phi} t + j \leq \mathcal{T}(\phi(0, 0)) \quad (1.11)$$

and

$$\lim_{(t,j) \in \text{dom } \phi: t+j \nearrow \mathcal{T}(\phi(0,0))} |\phi(t, j)|_{\mathcal{A}} = 0 \quad (1.12)$$

that does not require every maximal solution to be complete is given in [12, Definition 3.6 and Definition 7.1], which, to emphasize the potential lack of completeness, has the prefix “pre.”

This property becomes global when \mathcal{N} can be picked such that $\overline{C} \cup D \subset \mathcal{N}$. Condition (1.11) assures that convergence occurs at a point in $\text{dom } \phi$, in turn guaranteeing that the solution actually converges to \mathcal{A} .

The design of networked systems with such finite time convergence properties can be designed using the tools in [13], in particular, to guarantee network formation in finite time.

1.4.3 Forward invariance

A set K is said to be forward invariant for a dynamical system if every solution to the system from K stays in K for all future time. Also referred in the literature as flow-invariance and positively invariance, this property assure, the key property of interest in dynamical systems that solutions remain in a desired region of the state space. For a hybrid system \mathcal{H} on \mathbb{R}^n , a given set

$$K \subset \overline{C} \cup D$$

is said to be *forward pre-invariant* for \mathcal{H} if for each $x \in K$, each solution ϕ to \mathcal{H} with initial condition $\phi(0,0) = x$ is such that

$$\phi(t,j) \in K \quad \forall (t,j) \in \text{dom } \phi$$

The condition on K belonging to $\overline{C} \cup D$ is so that a set being pre-invariant is such that a solution exist from each point in it.⁵ The prefix “pre” indicates that the notion does not enforce that maximal solutions are complete, and when every maximal solution from K is complete, then the notion reduces to forward invariance as defined in [14] – see therein also “weak” notions of forward invariance.

Sufficient conditions guaranteeing forward pre-invariance of sets are given in [14, 15, 16] for general hybrid systems modeled as \mathcal{H} in (1.9). In particular, these conditions can be used as design tools to certify safety in a networked system.

1.4.4 Robustness

In real-world settings, networked systems are affected by a variety of perturbations that may compromise the satisfaction of the properties that they were designed for. Unmodeled dynamics in the models used for the agents

⁵ The solution might be *trivial* though, in the sense that its domain might be just one point – otherwise, points that are neither in \overline{C} nor in D would satisfy the invariance notion vacuously.

\mathcal{H}_i^a leads to perturbations of the data $(C_i^a, F_i^a, D_i^a, G_i^a, E_i^a, H_i^a)$. In particular, additive (in the general set-valued sense) perturbations to the flow map F_i^a and the jump map G_i^a can be used to capture terms that were omitted at the modeling stage, potentially with the intention of providing a simplified agent model that would enable analysis and design. Deflations and inflations of the sets C_i^a and D_i^a can be defined to model perturbations in the conditions allowing flows and jumps. Similar perturbations may appear in the models of the networks and algorithms. When such perturbations are not known at the design stage, one typically performs the design in nominal conditions, with the expectation that when the perturbations are present and have small size, then the established properties will hold, practically and semiglobally.

A semiglobal and practical (on the size of the perturbation) version of the asymptotic stability property of a set \mathcal{A} defined in Section 1.4.1 would guarantee, for every compact set $M \subset \mathbb{R}^n$ and every level of closeness $\epsilon > 0$, the existence of a maximum allowed perturbation size $\delta^* > 0$ such that every complete solution $\tilde{\phi}$ to \mathcal{H} under perturbations with size smaller than δ^* that has initial condition $\tilde{\phi}(0, 0) \in M$ is such that, in the limit as t or j grow unbounded, the distance from $\tilde{\phi}$ to \mathcal{A} is less than or equal to ϵ . When \mathcal{A} is an asymptotically stable compact set for \mathcal{H} , this property is guaranteed to hold under mild conditions on the data of \mathcal{H} . Such a result can be found in Chapter 7 of [12]; see Definition 7.18, Lemma 7.19, and Theorem 7.21 therein.

The property outlined above can be formally written in terms of a \mathcal{KL} bound. First, when \mathcal{H} is nominally well-posed and a compact set \mathcal{A} is asymptotically stable, then there exists a class- \mathcal{KL} function β such that every solution ϕ to \mathcal{H} satisfies

$$|\phi(t, j)|_{\mathcal{A}} \leq \beta(|\phi(0, 0)|_{\mathcal{A}}, t + j) \quad \forall (t, j) \in \text{dom } \phi \quad (1.13)$$

See [12, Theorem 7.12]. Then, [12, Theorem 7.21] implies that for each compact set $M \subset \mathbb{R}^n$ and every level of closeness $\epsilon > 0$, there exists $\delta^* > 0$ such that

$$|\tilde{\phi}(t, j)|_{\mathcal{A}} \leq \beta(|\phi(0, 0)|_{\mathcal{A}}, t + j) + \epsilon \quad \forall (t, j) \in \text{dom } \tilde{\phi} \quad (1.14)$$

for every solution $\tilde{\phi}$ that starts from M and that is under the effect of perturbations with size smaller than δ^* . A similar result for the case of finite time convergence is in Theorem 4.1 in [13].

Another typical perturbation in networked systems is the presence of noise in the quantities measured and transmitted by the network, and the values that finally arrive to the agents. When such noise is small, a semiglobal property that is practical on the size of the noise can be established using the tools mentioned above. When the noise is large, one is typically interested in characterizing the effect of the noise on the nominal asymptotic stability property, namely, on the distance to the set \mathcal{A} . The notion of input-to-state stability (ISS) is one way to characterize the effect of large noise. Denote

by $\tilde{\mathcal{H}}$ as the hybrid system under the effect of an exogenous disturbance d . The hybrid system $\tilde{\mathcal{H}}$ is input-to-state stable with respect to \mathcal{A} if there exist $\beta \in \mathcal{KL}$ and $\kappa \in \mathcal{K}$ such that each solution $\tilde{\phi}$ to $\tilde{\mathcal{H}}$ with associated disturbance d satisfies

$$|\tilde{\phi}(t, j)|_{\mathcal{A}} \leq \max\{\beta(|\phi(0, 0)|_{\mathcal{A}}, t + j), \kappa(\|d\|_{(t, j)})\} \quad (1.15)$$

for each⁶ $(t, j) \in \text{dom } \tilde{\phi}$. Several characterizations and Lyapunov-based tools to certify ISS in hybrid systems are given in [17]. The set \mathcal{KL} is the set of \mathcal{KL} functions and \mathcal{K} is the set of \mathcal{K} functions; see [12, Section 3.5].

A direct approach to design algorithms conferring robustness is to perform the design task using a model that explicitly includes a model of the perturbations. Such an approach allows for a variety of perturbations, as long as they can be modeled and a certificate guaranteeing the desired properties can be found. When robust asymptotic stability is of interest, robust control Lyapunov functions for hybrid systems can be employed to guarantee robust asymptotic stability of sets; see [18]. Forward invariance of sets with robustness to perturbations can be certified for hybrid dynamical systems when the model includes the perturbations. Tools for the design of algorithms conferring robust forward invariance to general sets and, in particular, to sets given by sublevel sets of Lyapunov functions are available in [16].

Delay is a perturbation that is of particular interest in networked systems as it is unavoidable in real-world settings. Compared to the tools to deal with the sources of perturbations mentioned above, design methods to guarantee robustness to delays are much less developed. Works pertaining to systems with hybrid dynamics and delays have focused on guaranteeing pre-asymptotic stability through the use of Razumikhin functions [19, 20] and Lyapunov functionals for retarded functional impulse differential equations [21]. Results for switched systems with delays are also available in [22, 23, 24, 25]. Results for linear reset systems with delays developed using passivity appeared in [26, 27]. Tools for the study of delays in hybrid systems modeled as in (1.9) have recently appeared in the sequence of articles [28, 29, 30, 31], which provide tools to study the effects of general delays. Along a different vein, in [32], we have recently proposed a way to exploit well-posedness and a \mathcal{KL} bound as in (1.13) to handle the sole effect of delays on events in hybrid systems.

⁶ A pair $(\tilde{\phi}, d)$ defines a solution to $\tilde{\mathcal{H}}$ if it satisfies its dynamics. Given a hybrid arc d , its sup norm at $(t, j) \in \text{dom } d$ is

$$\|d\|_{(t, j)} := \max \left\{ \text{ess sup}_{(s, k) \in \text{dom } d \setminus \Gamma(d), s+k \leq t+j} |d(s, k)|, \sup_{(s, k) \in \Gamma(d), s+k \leq t+j} |d(s, k)| \right\}$$

where $\Gamma(d)$ denotes the set of all $(t, j) \in \text{dom } d$ such that $(t, j + 1) \in \text{dom } d$.

1.5 Applications

The models and tools presented in the previous sections have recently been used to solve problems pertaining to certain classes of networked hybrid dynamical systems. In [1, 2], a distributed hybrid observer to estimate the state of a linear time-invariant system was designed to guarantee asymptotic stability of a set on which the state estimation error is zero. In [3], a solution to the control problem of steering the state of the agents with point-mass dynamics to the same value over a network that only allows exchange of information at isolated, aperiodic time instances is proposed. The algorithm discretely updates the input to the point-mass system at communication events and, in between events, changes continuously according to a linear differential equation. A hybrid control algorithm for the synchronization of multiple systems with general linear time-invariant dynamics over a similar communication network appeared in [33, 7]. The remainder of this section presents a summary of these results.

1.5.1 *Distributed Estimation*

State estimation in networked systems has seen increased attention recently. These include continuous-time algorithms for distributed estimation of the state of a plant in [34] with robustness guarantees and in [35, 36], both when information is exchanged continuously. Algorithms for which information arrives at common discrete time instances include the network of local observers proposed in [37] for linear time-invariant plants and the optimal estimators in [38], for time-varying networked systems, both in discrete time and with information shared at each discrete time instant. Approaches that keep the continuous dynamics of the plant and treat the communication events as impulsive events include the observer-based controller [39] for network control systems modeled as time-varying hybrid systems, the observer-protocol pair in [40] to asymptotically reconstruct the state of a linear time-invariant plant using periodic measurements from multiple sensors, the distributed observer in [41] designed by partitioning the dynamics into disjoint areas and attaching an algorithm to each area that updates the estimates over time windows with common length, and the robust continuous-time observer for estimation in network control systems in [42] designed via an emulation-like approach and exploiting trajectory-based and small-gain arguments. Other approaches that mix continuous and discrete dynamics have appeared in the nonlinear and stochastic systems literature; see [43, 44, 45, 46, 47, 48, 49, 50, 51].

In this section, we consider the problem of estimating the state of a dynamical system from intermittent measurements of functions of its output over a network with N' nodes, each running a decentralized state estimator. The communication events occur according to one of the models in Exam-

ple 4. Under nominal conditions, the model governing the dynamics of the system to estimate the state of is given by a linear time-invariant system. The algorithm we propose builds from the hybrid observer in [52], which is shown to guarantee global exponential stability of the zero-estimation error under sporadic measurements. Without loss of generality, following the model in (1.2) and defining $N = N' + 1$, we assume that the first agent corresponds to this dynamical system, while the dynamics of agents with $i \in \mathcal{V}' := \{2, 3, \dots, N' + 1\}$ implement the decentralized state estimators. In this way, the dynamics of the first agent are given by

$$\dot{z}_1 = Az_1 \quad (1.16)$$

where $z_1 \in \mathbb{R}^{n_1^a}$ denotes its state and $A \in \mathbb{R}^{n_1^a \times n_1^a}$ is the system matrix. For each $i \in \mathcal{V}'$, the i -th agent running a state estimator receives the measurement

$$y_i = H_i z_1 \quad (1.17)$$

and the outputs y_k of its neighbors, that is, for each $k \in \mathcal{N}(i)$, at time instances t_j^i satisfying

$$t_{j+1}^i - t_j^i \in [T_1^i, T_2^i], \quad j > 0 \quad (1.18)$$

where $H_i \in \mathbb{R}^{p_i^a \times n_i^a}$ is the local output matrix of the i -th agent and $T_2^i \geq T_1^i > 0$ are parameters that, as T_2 and T_1 in (1.5), determine the minimum and maximum amount of time to elapse between communication events for the i -th agent. Following the network models proposed in Section 1.2.2, in particular, those in Example 4, we employ the model

$$\dot{\tau}_i = 1 \quad \tau_i \in [0, T_2^i] \quad (1.19a)$$

$$\tau_i^+ \in [T_1^i, T_2^i] \quad \tau_i = 0 \quad (1.19b)$$

to trigger the events at which the i -th agent receives y_i and the y_k 's. Since the information from all neighbors to agent i arrives simultaneously, we can employ a single state μ_i for each agent, rather than d_i^{in} states μ_{ik} for each agent, $i \in \mathcal{V}'$. A model as in (1.3) can be derived following the construction in Example 4, where the state μ_i would be given by τ_i and the input ω_i by the information to transmit, namely, y_i and the y_k 's.

We propose a decentralized hybrid algorithm that, at each agent and by employing information received from the neighbors over a communication graph, generates a converging estimate of the state of the first agent. More precisely, at the i -th agent, $i \in \mathcal{V}'$, the hybrid algorithm has a state with a variable $\hat{z}_i \in \mathbb{R}^{n_1^a}$ storing the estimate of the state z_1 and an information fusion state variable, denoted ℓ_i , storing the measurements received from its neighbors. These state variables are continuously updated by a differential equations

$$\dot{\hat{z}}_i = A\hat{z}_i + \ell_i \quad (1.20a)$$

$$\dot{\ell}_i = h_i \ell_i \quad (1.20b)$$

when no information is received, while when information is received, the states \hat{z}_i and ℓ_i are updated according to

$$\hat{z}_i^+ = \hat{z}_i \quad (1.21a)$$

$$\ell_i^+ = \sum_{k \in \mathcal{N}(i)} G_{oi}^k(\hat{z}_i, \hat{z}_k, y_i, y_k) \quad (1.21b)$$

with

$$G_{oi}^k(\hat{z}_i, \hat{z}_k, y_i, y_k) = \frac{1}{d_{in}^i} K_{ii} y_i^e + K_{ik} y_k^e + \gamma(\hat{z}_i - \hat{z}_k) \quad (1.22)$$

where, for each $i, k \in \mathcal{V}'$, $y_i^e = H_i \hat{z}_i - y_i$ is the output estimation error; the scalars h_i and γ , and the matrix K_{ik} define the parameters of the algorithm. The constants g_{ik} in (1.21) and d_{in}^i in (1.22) are associated with the communication graph, which is assumed to be given. The map G_{oi}^k defines the impulsive update law when new information is collected from the first agent and the k -th neighbor for agent i . The information fusion state ℓ_i is injected into the continuous dynamics of the local estimate \hat{z}_i and, at communication events, injects new information impulsively – the right-hand side of (1.21) is the “innovation term” of the proposed observer. The specific update law in (1.22) is such that the second term in (1.22) uses the output error of each k -th agent that is a neighborhood of the i -th agent, and the third term in (1.22) uses the difference between the estimates \hat{z}_i and \hat{z}_k . These are the quantities that are transmitted (instantaneously) at communication events only.

The continuous and discrete dynamics in (1.20) and (1.21) can be modeled as a hybrid algorithm \mathcal{H}_i^K as in (1.8) with state $\eta_i = (\hat{z}_i, \ell_i)$, input $v_i = (y_i, \{(\hat{z}_k, y_k)\}_{k \in \mathcal{N}(i)}, \mu_i)$, and data given by

$$\begin{aligned} F_i^K(\eta_i, v_i) &:= \begin{bmatrix} A\hat{z}_i + \ell_i \\ h_i \ell_i \end{bmatrix} \\ C_i^K &:= \{(\eta_i, v_i) : \mu_i \in [0, T_2]\} \\ G_i^K(\eta_i, v_i) &:= \left[\sum_{k \in \mathcal{N}(i)} \frac{1}{d_{in}^i} K_{ii} (H_i \hat{z}_i - y_i) + K_{ik} (H_k \hat{z}_k - y_k) + \gamma(\hat{z}_i - \hat{z}_k) \right] \\ D_i^K &:= \{(\eta_i, v_i) : \mu_i = 0\} \\ H_i^K(\eta_i, v_i) &:= \hat{z}_i \end{aligned}$$

and E_i^K the entire state and input space of \mathcal{H}_i^K . Note that the input to the algorithm includes the output $\mu_i = x_i$ of the network model in (1.7). Due to μ_i triggering the jumps in \mathcal{H}_i^K , for each $i \in \mathcal{V}'$, jumps of the network and the hybrid algorithm for the i -th agent occur simultaneously.

The goal of the algorithm is to guarantee that, for each $i \in \mathcal{V}'$, the estimate \hat{z}_i converges to the state z_1 . When the estimates are equal to z_1 , the update law maps ℓ_i to zero. Noting that the timers $\tau_i (= \mu_i)$ in the model of the network remain within the set $[0, T_2^i]$, the goal of the algorithm is to render the set

$$\mathcal{A} := \{x : z_1 = \hat{z}_i, \mu_i \in [0, T_2^i], \ell_i = 0 \quad \forall i \in \mathcal{V}'\} \quad (1.23)$$

for the resulting closed-loop system with state x , which is given by the stack of the state variables of the first agent (z_1), each algorithm (η_i 's), and each network (μ_i 's).

A result for the design of the parameters of the proposed hybrid algorithm can be found in [2] to guarantee that the set \mathcal{A} in (1.23) is globally exponentially stable for the closed-loop hybrid system \mathcal{H} . Given the network parameters $0 < T_1^i \leq T_2^i$ for each $i \in \mathcal{V}'$, it is assumed that the N' agents are connected via a digraph $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{G})$ that is such that there exist a constant $\delta > 0$ and matrices $K_g, P = P^\top > 0, Q_i = Q_i^\top > 0$ satisfying⁷

$$\mathcal{M}(\tau) := \begin{bmatrix} \text{He}(A_\theta, P) & -P + \tilde{A}_\theta^\top \mathcal{K}^\top \tilde{Q}(\tau) \\ \star & -\delta \tilde{Q}(\tau) - \text{He}(\tilde{\mathcal{K}}, \tilde{Q}(\tau)) \end{bmatrix} < 0 \quad \forall \tau \in \mathcal{T} \quad (1.24)$$

where $\tau = (\tau_2, \tau_3, \dots, \tau_N)$, $\mathcal{T} := [0, T_2^2] \times [0, T_2^3] \times \dots \times [0, T_2^N]$,

$$\begin{aligned} A_\theta &= I_N \otimes A + \mathcal{K} \\ \mathcal{K} &= (K_g H_g) * (I_N + \mathcal{G}) + \gamma \mathcal{L} \otimes I_n \\ H_g &= \text{diag}(H_2, H_3, \dots, H_N) \\ \tilde{A}_\theta &= A_\theta - \tilde{H} \\ \tilde{\mathcal{K}} &= \mathcal{K} - \tilde{H} \\ \tilde{H} &= \text{diag}(h_2 I_n, h_3 I_n, \dots, h_N I_n) \\ \tilde{Q}(\tau) &= \text{diag}(\tilde{Q}_2(\tau_2), \tilde{Q}_3(\tau_3), \dots, \tilde{Q}_N(\tau_N)) \\ \tilde{Q}_i(\tau_i) &= \exp(\delta \tau_i) Q_i \end{aligned}$$

These design conditions are obtained using sufficient conditions for asymptotic stability in [53] (specifically, Proposition 3.29 therein), which for the current data turns out to be exponential, and a convenient change of coordinates. The Lyapunov function used to show global exponential stability of the set \mathcal{A} in (1.23) is given by

$$V(x) := e^\top P e + \theta^\top \tilde{Q}(\tau) \theta$$

where $e = (e_2, e_3, \dots, e_N)$, $e_i = \hat{z}_i - z_1$, $\tau = (\tau_2, \tau_3, \dots, \tau_N)$, $\theta = (\theta_2, \theta_3, \dots, \theta_N)$, and

⁷ Given matrices A and B , $\text{He}(A, B) = A^\top B + B^\top A$, $A \otimes B$ defines the Kronecker product, and $A * B$ the Khatri-Rao product. The matrix I_n is the $n \times n$ identity matrix.

$$\theta_i = K_{ii}y_i^e + \sum_{k \in \mathcal{N}(i)} K_{ik}y_k^e + \gamma \sum_{k \in \mathcal{N}(i)} (\hat{z}_i - \hat{z}_k) - \ell_i \quad (1.25)$$

for each $i \in \mathcal{V}'$, with P and \tilde{Q} as defined above. Note that $V(x) = 0$ for each $x \in \mathcal{A}$, while for any $x \notin \mathcal{A}$, $V(x)$ is positive. More importantly, intuitively, regardless of which timer triggers a jump, this function satisfies the useful property that $V(x^+) - V(x)$ is upper bounded by a nonpositive function of θ_i for all x in the jump set. Such a property is possible due to the convenient choice of the update law of the observer used at jumps, which, in the coordinates in (1.25), leads to e being mapped by the identity and θ_i to zero. The injection of ℓ_i in the flows of the local estimate in (1.20) and the continuous dynamics of ℓ_i further permit a decrease of V during flows, which conveniently uses exponential functions in the definition of \tilde{Q} . These properties are exploited to arrive to the result above. The interested reader is referred to [2], where in addition to several other results pertaining to design, nominal and ISS-type robustness of the above algorithm, several examples are provided.

1.5.2 Intermittent Consensus

Consensus emerges in a variety of applications, such as spiking neurons, formation control and flocking maneuvers [54, 55, 56, 57], distributed sensor networks [58, 59], satellite constellation formation [60], and in communication of computer network systems [61]. Consensus for continuous-time and discrete-time systems has been widely studied in the literature; see, e.g., [55] and [62]. On the other hand, the sometimes unavoidable limitation that information is available only at intermittent time instances imposes a challenge to the design of such algorithms, as much of the literature that departs from the standard settings reveal [63, 64, 65, 66, 67]. In this section, we consider the problem of steering the state of a group of N agents with dynamics

$$\dot{z}_i = u_i \quad i \in \mathcal{V} \quad (1.26)$$

to consensus under intermittent information constraints over a digraph $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{G})$, where $z_i \in \mathbb{R}$ is the state and $u_i \in \mathbb{R}$ is the control input of the i -th agent. The i -th agent receives the states of its neighbors at the events triggered by network model as in Example 4. As in Section 1.5.1, we attach a single event to each agent and employ the communication model given in (1.7) with common minimum and maximum time in between events, i.e., $T_1^i = T_1$ and $T_2^i = T_2$ for each $i \in \mathcal{V}$. Following [55, 56], the goal is to design a decentralized algorithm assigning the input u_i of each agent so that every resulting maximal solution ϕ is complete and its $z = (z_1, z_2, \dots, z_N)$ component ϕ_z satisfies

$$\lim_{t+j \rightarrow \infty} |\phi_{z_i}(t, j) - \phi_{z_k}(t, j)| = 0$$

for each $i, k \in \mathcal{V}$, namely, global attractive synchronization with respect to z , as defined in Section 1.3.2.

Due to the aperiodic and impulsive arrival of information, analysis tools that have been shown useful for continuous-time or discrete-time consensus algorithms do not apply for the design of an algorithm accomplishing the stated goal. Motivated by this, we solve this problem with a hybrid control algorithm that assigns u_i to a variable in the controller, which is denoted ℓ_i , that at communication events collects information about its states from its neighbors, computes a local quantity determining how far apart the received state information is relative to its own state, and, in between events, changes according to a linear differential equation. For each $i \in \mathcal{V}$, given the parameter T_2 of the network model, the i -th hybrid controller has state ℓ_i with the following dynamics:

$$\begin{aligned} u_i &= \ell_i \mu_i \\ \dot{\ell}_i &= h \ell_i & \mu_i &\in [0, T_2] \\ \ell_i^+ &= -\gamma \sum_{k \in \mathcal{N}(i)} (z_i - z_k) & \mu_i &= 0 \end{aligned} \quad (1.27)$$

where γ and h are parameters of the algorithm. The continuous and discrete dynamics in (1.27) can be modeled as a hybrid algorithm \mathcal{H}_i^K as in (1.8) with state $\eta_i = \ell_i$, input $v_i = (z_i, \{z_k\}_{k \in \mathcal{N}(i)}, \mu_i)$, and data given by

$$\begin{aligned} F_i^K(\eta_i, v_i) &:= h \ell_i \\ C_i^K &:= \{(\eta_i, v_i) : \mu_i \in [0, T_2]\} \\ G_i^K(\eta_i, v_i) &:= -\gamma \sum_{k \in \mathcal{N}(i)} (z_i - z_k) \\ D_i^K &:= \{(\eta_i, v_i) : \mu_i = 0\} \\ H_i^K(\eta_i, v_i) &:= \eta_i \end{aligned}$$

and E_i^K the entire state and input space of \mathcal{H}_i^K .

The resulting hybrid system from interconnecting the plant model in (1.26), the network model in (1.7), and the algorithm model in (1.27) has state $x = (z, \ell, \tau)$, where $z = (z_1, z_2, \dots, z_N)$, $\ell = (\ell_1, \ell_2, \dots, \ell_N)$, and $\mu = (\mu_1, \mu_2, \dots, \mu_N)$. The hybrid system \mathcal{H} is then given by

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \ell \\ 0 \\ -\mathbf{1} \end{bmatrix} =: F(x) & x \in C := \mathbb{R}^N \times \mathbb{R}^N \times \mathcal{T}, \\ x^+ &\in G(x) & x \in D := \bigcup_{i \in \mathcal{V}} D_i \end{aligned} \quad (1.28)$$

where $\mathbf{1}$ is the column vector of dimension N and all entries equal to 1,

$$D_i := \{x \in \mathbb{R}^N \times \mathbb{R}^N \times \mathcal{T} : \mu_i = 0, \mu_k > 0 \quad \forall k \in \mathcal{V} \setminus \{i\}\}$$

G_i is defined as

$$G_i(x) := (z, \eta_1, \eta_2, \dots, \eta_{i-1}, G_i^K(\eta_i, v_i), \eta_{i+1}, \dots, \eta_N, \mu_1, \mu_2, \dots, \mu_{i-1}, [T_1, T_2], \mu_{i+1}, \dots, \mu_N)$$

for each $x \in D_i$, and otherwise as the empty set, and $\mathcal{T} = [0, T_2]^N$. The definition of G_i is such that only the i -th component of η and of μ are updated, where $i \in \mathcal{V}$.

Due to the fact that the timer variable μ_i being zero is the only trigger of the jumps, some properties of the domain of solutions can easily be characterized. In particular, a solution ϕ to the hybrid system \mathcal{H} is such that every $(t, j) \in \text{dom } \phi$ satisfies

$$(j-1)T_1 \leq t \leq (j+1)T_2 \quad \forall j \geq 1$$

Moreover, due to the assumption that $T_1 > 0$, every maximal solution ϕ is complete and the hybrid time domain is unbounded in both t and j .

The decentralized hybrid control algorithm proposed above can be designed to guarantee the desired consensus property with an exponential decay rate and, furthermore, to render Lyapunov stable the set

$$\mathcal{A} := \{x : z_i = z_k = \bar{z}, \ell_i = 0, \mu_i \in [0, T_2] \quad \forall i, k \in \mathcal{V}\} \quad (1.29)$$

for some constant \bar{z} . Such a result appeared in [3]. Given two positive scalars such that $T_1 \leq T_2$ and a strongly connected and weight balanced digraph Γ , if there exist scalars $\gamma, h \in \mathbb{R}$, and $\sigma > 0$, positive definite diagonal matrices P and Q such that

$$\begin{bmatrix} \gamma \text{He}(P, \mathcal{L}) & -P\Pi + \mathcal{K}_1^\top Q E(\tau) \\ \star & -\sigma Q E(\tau) - \text{He}(Q E(\tau), \mathcal{K}_2) \end{bmatrix} \leq 0 \quad \forall \tau \in \mathcal{T} \quad (1.30)$$

where $\Pi = I - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^\top$, $\mathcal{K}_1 = \gamma \mathcal{K}_2 \mathcal{L}$, $\mathcal{K}_2 = \gamma \mathcal{L} - hI$, and

$$E(\tau) = \text{diag}(\exp(\sigma\tau_1), \exp(\sigma\tau_2), \dots, \exp(\sigma\tau_N))$$

then the set \mathcal{A} in (1.29) is globally exponentially stable (GES) for the hybrid system \mathcal{H} in (1.28).

The case when all of the agents exchange information intermittently, but at the same time instances, can be modeled using a single timer. In such a case, the timer state of the model in (1.28) reduces to the single timer $\mu \in [0, T_2]$, \mathcal{T} to $[0, T_2]$, the jump set to $D = \{x \in \mathbb{R}^N \times \mathbb{R}^N \times \mathcal{T} : \mu = 0\}$, and the jump map to $G(x) = (z, G_1^K(\eta_1, v_1), G_2^K(\eta_2, v_2), \dots, G_N^K(\eta_N, v_N), [T_1, T_2])$. Furthermore, the sufficient condition in the result above can be further sim-

plified. It can be shown that the set \mathcal{A} in (1.29) is GES for the hybrid system in (1.28) with a single timer if either one of the following properties hold:

- (1) the digraph is strongly connected, and there exist a positive scalar γ and a positive definite symmetric matrix P satisfying

$$A_g^\top \exp(A_f^\top \tau) P \exp(A_f \tau) A_g - P < 0 \quad \forall \tau \in [T_1, T_2], \quad (1.31)$$

with $A_f = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$, $A_g = \begin{bmatrix} I & 0 \\ -\gamma \bar{\mathcal{L}} & 0 \end{bmatrix}$, and $\bar{\mathcal{L}}$ is a diagonal matrix with diagonal elements $(\lambda_2, \lambda_3, \dots, \lambda_N)$ with λ_i 's being the positive eigenvalues of \mathcal{L} .

- (2) the digraph is completely connected, and there exist a positive scalar γ and a positive definite symmetric matrix P satisfying

$$A_g^\top \exp(A_f^\top \tau) P \exp(A_f \tau) A_g - P < 0 \quad \forall \tau \in [T_1, T_2], \quad (1.32)$$

with $A_f = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $A_g = \begin{bmatrix} 1 & 0 \\ -\gamma & 0 \end{bmatrix}$.

Note that, as a difference to (1), the condition in (2) does not involve the graph. This is possible due to the assumption therein about the graph being completely connected.

Interestingly, if in addition to either one of the conditions above the digraph Γ is weight balanced, then every maximal solution $\phi = (\phi_z, \phi_\ell, \phi_\mu)$ satisfies $\lim_{t+j \rightarrow \infty} \phi_\ell(t, j) = 0$ and

$$\lim_{t+j \rightarrow \infty} \phi_z(t, j) = \rho(\phi(0, 0)) =: \bar{z}$$

where

$$\rho(\phi(0, 0)) := \left(\frac{1}{N} (\mathbf{1}_N^\top \phi_x(0, 0) + \mathbf{1}_N^\top \phi_\ell(0, 0) \phi_\mu(0, 0)) \right) \mathbf{1}_N. \quad (1.33)$$

This property not only characterizes the value of \bar{z} in the set \mathcal{A} in (1.29) but also implies that \mathcal{A} is pointwise exponentially stable.

The conditions for the design of the proposed consensus algorithm were obtained using Lyapunov functions for the entire networked hybrid system. The details can be found in [3]. In particular, the key idea to derive the conditions for the single timer case is to exploit properties of the Laplacian matrix for a strongly connected digraph Γ . For such graphs there exists a nonsingular matrix $T = [\mathbf{1}_N, T_1]$ such that $T^{-1} \mathcal{L} T = \begin{bmatrix} 0 & 0 \\ 0 & \bar{\mathcal{L}} \end{bmatrix}$, which is a diagonal matrix containing the eigenvalues of \mathcal{L} , where $\bar{\mathcal{L}}$ is a diagonal matrix with diagonal elements $(\lambda_2, \lambda_3, \dots, \lambda_N)$ with λ_i 's being the positive eigenvalues of \mathcal{L} . This property leads to the change the coordinates $\tilde{z} = T^{-1} z$ and $\tilde{\ell} = T^{-1} \ell$, and the resulting dynamics

$$\begin{aligned}\dot{\tilde{z}} &= T^{-1}\dot{z} = T^{-1}\ell = \tilde{\ell} \\ \dot{\tilde{\ell}} &= T^{-1}\dot{\ell} = 0.\end{aligned}$$

during flows, and

$$\begin{aligned}\tilde{z}^+ &= T^{-1}z^+ = T^{-1}z = \tilde{z} \\ \tilde{\ell}^+ &= -\gamma T^{-1}\mathcal{L}z = -\gamma T^{-1}\mathcal{L}T\tilde{z} = -\gamma \begin{bmatrix} 0 & 0 \\ 0 & \bar{\mathcal{L}} \end{bmatrix} \tilde{z}\end{aligned}$$

at jumps. These new coordinates are convenient as they allow to analyze the evolution of the scalar states \tilde{z}_1 and $\tilde{\ell}_1$ separately from the remaining state components of \tilde{z} and $\tilde{\ell}$.

It can be shown that, for the case of a single timer, the consensus set is partially pointwise globally exponentially stable, which is a stronger notion than typical notions of asymptotic stability due to the additional requirement that each point in the set is Lyapunov stable; see e.g. [68] and [69]. Furthermore, it can be shown that the consensus property is robust to a class of perturbations on the information. See [3].

1.5.3 Distributed Synchronization

Synchronization is a property of interest in many problems emerging in science and engineering, such as spiking neurons [70, 5], formation control and flocking [54, 56], distributed sensor networks [59], and satellite constellation formation [60], among others. The literature about synchronization is quite rich, with numerous contributions employing a variety of techniques, such as Lyapunov functions [71, 72], convergence [73, 74], contraction theory [75], and incremental input-to-state stability [76, 77]. Synchronization for continuous-time systems where communication coupling occurs at discrete events is an emergent area of study. In [77], the authors study a case of synchronization where agents have nonlinear continuous-time dynamics with continuous coupling and impulsive perturbations. In [78], the authors use Lyapunov-like analysis to derive sufficient conditions for the synchronization of continuously coupled nonlinear systems with impulsive resets on the difference between neighboring agents. In [79], a distributed event-triggered control strategy was developed to drive the outputs of the agents in a network to synchronization. Using a sample-and-hold self-triggered controller policy, a practical synchronization result was established in [80] for the case of first-order integrator dynamics. On the other hand, methods for the design of algorithms that guarantee synchronization of multi-agent systems with information arriving at impulsive, asynchronous time instances are not available.

In this section, we consider the problem of synchronizing the state of N networked agents from intermittent measurements of the state (or of a func-

tion of it) over a digraph. Each agent runs a decentralized hybrid algorithm that uses information received from its neighbors. The nominal model of the agents is given as follows: for each $i \in \mathcal{V}$,

$$\dot{z}_i = Az_i + Bu_i \quad (1.34)$$

where A is the nominal system matrix and B is the input matrix. The i -th agent in the network measures its local output, denoted y_i , and the information received from its neighbors, denoted y_k , at the communication events, where

$$y_i = Hz_i \quad (1.35)$$

with H being the output matrix. Following the network models proposed in Section 1.2.2, in particular, those in Example 4, we employ the model in (1.7) to trigger the events at which the i -th agent receives the y_k 's.

To globally synchronize the states of the N agents, we propose the following decentralized hybrid algorithm for each $i \in \mathcal{V}$: the algorithm has a memory state, denoted ℓ_i , that when information arrives, is updated to the relative error between the output of the i -th agent and those received from its neighbors, namely,

$$\ell_i^+ = K \sum_{k \in \mathcal{N}(i)} (y_i - y_k) = KH \sum_{k \in \mathcal{N}(i)} (z_i - z_k) \quad (1.36)$$

where K is a constant matrix to be designed, and in between communication events is continuously updated according to

$$\dot{\ell}_i = M\ell_i \quad (1.37)$$

where M is a constant matrix to be designed. Following the construction of the hybrid algorithms in Section 1.5.1 and in Section 1.5.2, this algorithm can be modeled as \mathcal{H}_i^K in (1.8) with state $\eta_i = \ell_i$, input $v_i = (y_i, \{y_k\}_{k \in \mathcal{N}(i)}, \mu_i)$, and data given by

$$F_i^K(\eta_i, v_i) := M\ell_i \quad (1.38)$$

$$C_i^K := \{(\eta_i, v_i) : \mu_i \in [0, T_2^i]\} \quad (1.39)$$

$$G_i^K(\eta_i, v_i) := KH \sum_{k \in \mathcal{N}(i)} (z_i - z_k) \quad (1.40)$$

$$D_i^K = \{(\eta_i, v_i) : \mu_i = 0\} \quad (1.41)$$

$$H_i^K(\eta_i, v_i) := \ell_i \quad (1.42)$$

and E_i^K the entire state and input space of \mathcal{H}_i^K . Also, note that the input to the algorithm includes the output μ_i of the network model in (1.7), leading to

jumps of the network and the hybrid algorithm for the i -th agent occurring simultaneously.

The goal of the synchronization algorithm introduced above is to guarantee that, for each $i, k \in \mathcal{V}$, the error between z_i and z_k converges to zero, with stability. These requirements correspond to the notions of stable and attractive synchronization introduced in Section 1.3.2. When the estimates are equal to z_1 , the update law maps ℓ_i to zero. Noting that when the states of all of the agents coincide we have that the ℓ_i 's are reset to zero and that the timers τ_i in the model of the network remain within the set $[0, T_2^i]$, the goal of the algorithm is to render the set

$$\mathcal{A} := \{x : z_i = z_k \ \forall i, k \in \mathcal{V}, \mu_i \in [0, T_2^i], \ell_i = 0 \ \forall i \in \mathcal{V}\} \quad (1.43)$$

globally asymptotically stable for the resulting closed-loop system \mathcal{H} with state x , which is given by the stack of the state variables of each agent (z_i), each algorithm (η_i 's), and each network (μ_i 's).

Results for the design of the parameters M and K of the proposed hybrid algorithm can be found in [33] to guarantee that the set \mathcal{A} in (1.43) is globally exponentially stable for \mathcal{H} , and hence, global exponential synchronization is achieved. Given the network parameters $0 < T_1^i \leq T_2^i$ for each $i \in \mathcal{V}$ and a undirected graph Γ , the set \mathcal{A} in (1.43) is globally exponentially stable for the hybrid closed-loop system \mathcal{H} resulting from controlling the agents in (1.34) with hybrid algorithms as in (1.38)-(1.42) over a network modeled as in (1.7) if there exist scalars $\sigma > 0$, $\varepsilon \in (0, 1)$, matrices K and M , and positive definite symmetric matrices P_i, Q_i for each $i \in \mathcal{V}'$, satisfying

$$\mathcal{M}(\nu) := \begin{bmatrix} \text{He}(P, \bar{A}) - P\bar{B} + \exp(\sigma\nu)(\bar{K}\bar{A} - \bar{M}\bar{K})^\top Q \\ \star & \text{He}(\exp(\sigma\nu)Q, \bar{M} - \bar{K}\bar{B} - \frac{\sigma}{2}I) \end{bmatrix} < 0 \quad \forall \nu \in [0, \bar{T}] \quad (1.44)$$

where $\bar{A} = I \otimes A + \Lambda \otimes BKH$, $\bar{B} = I \otimes B$, $\bar{M} = I \otimes M$, $\bar{K} = \Lambda \otimes KH$, $\Lambda = \text{diag}(\lambda_2, \lambda_3, \dots, \lambda_N)$ where λ_i are the nonzero eigenvalues of \mathcal{L} , and

$$(1 - \varepsilon)\underline{T} - \frac{\alpha_2 \sigma \bar{T}}{\beta} > 0. \quad (1.45)$$

where $\underline{T} := \min_{i \in \mathcal{V}} T_1^i$, $\bar{T} := \max_{i \in \mathcal{V}} T_2^i$,

$$\begin{aligned} \beta &= - \max_{\nu \in [0, \bar{T}]} \bar{\lambda}(\mathcal{M}(\nu)) \\ \alpha_2 &= \max\{\bar{\lambda}(P), \bar{\lambda}(Q) \exp(\sigma \bar{T})\}. \end{aligned}$$

Moreover, every maximal solution ϕ to the closed-loop system satisfies

$$|\phi(t, j)|_{\mathcal{A}} \leq \kappa \exp(-r(t + j)) |\phi(0, 0)|_{\mathcal{A}} \quad \forall (t, j) \in \text{dom } \phi \quad (1.46)$$

where $\kappa = \sqrt{\frac{\alpha_2}{\alpha_1}} \exp\left(\frac{\beta(1-\varepsilon)\underline{T}}{2\alpha_2}\right)$ and $r = \frac{\beta}{2\alpha_2 N} \min\left\{\varepsilon N, (1-\varepsilon)\underline{T} - \frac{\alpha_2\sigma\bar{T}}{\beta}\right\}$, and $\alpha_1 = \min\{\underline{\lambda}(P), \underline{\lambda}(Q)\}$.

To arrive to these design conditions, we employed the property that

$$\theta_i = KH \sum_{k \in \mathcal{N}(i)} (z_i - z_k) - \ell_i. \quad (1.47)$$

is reset to zero at jumps due to the timer τ_i expiring alone. It follows that the quantity

$$V(x) = \begin{bmatrix} z \\ \theta \end{bmatrix}^\top \bar{\Psi} R(\tau) \bar{\Psi}^\top \begin{bmatrix} z \\ \theta \end{bmatrix}, \quad (1.48)$$

with⁸ $\bar{\Psi} = \text{diag}(\tilde{\Psi} \otimes I_n, \tilde{\Psi} \otimes I_p)$, where $\tilde{\Psi} = (\psi_2, \psi_3, \dots, \psi_N) \in \mathbb{R}^{N \times N-1}$, $\psi_i = (\psi_{i1}, \psi_{i2}, \dots, \psi_{iN})$ being the orthonormal eigenvector corresponding to the nonzero eigenvalue λ_i of \mathcal{L} , $i \in \mathcal{V}$ (furthermore, $\sum_{k=1}^N \psi_{ik} = 0$), $R(\tau) = \text{diag}(P, Q \exp(\sigma\bar{\tau}))$, $\bar{\tau} = \frac{1}{N} \sum_{i=1}^N \tau_i$, $P = \text{diag}(P_2, P_3, \dots, P_N)$, and $Q = \text{diag}(Q_2, Q_3, \dots, Q_N)$, decreases during flows due to (1.44), while at jumps, its potential growth can be dominated by imposing (1.45); cf. the construction of the Lyapunov function in Section 1.5.1, where such a Lyapunov function decreases during flows and has a nonpositive change at jumps. To guarantee exponential stability of the synchronization set, the result [53, Proposition 3.29], which uses a balancing condition between jumps and flows to guarantee that solutions converge to the desired set, exploited.

1.6 Final Remarks and Acknowledgments

Hybrid systems models, along with their associated notions and tools, lead to powerful methods for the design of algorithms conferring desired dynamical properties in complex networks. The methods summarized in this book chapter are suitable for settings in which the combination of continuous and discrete behavior is unavoidable, digital networks govern the exchange of information between the agents, information is limited and with uncertainty, and the algorithms are distributed. The proposed networked hybrid systems framework allows for hybrid models at the agent, network, and algorithm level. The applications of the notions and tools to estimation, consensus, and synchronization over networks are just examples of the power of the hybrid systems framework, being the hope that they will inspire the formulation of new notions and tools suitable for networked hybrid systems as well as the solution to challenging applications.

⁸ A digraph is undirected if and only if the Laplacian is symmetric. The construction of $\tilde{\Psi}$ is inspired by [81].

I would like to acknowledge and thank my collaborators who have contributed to the ideas presented in this book chapter. Part of the work presented here was done in collaboration with my PhD students Yuchun Li and Sean Phillips, who, respectively, have lead our research on distributed estimation and distributed synchronization using hybrid systems methods. The distributed estimation strategy and the nondeterministic network model using timers was developed with Francesco Ferrante, Frederic Gouaisbaut, and Sophie Tarbouriech. The formulation of the safety notion was inspired by work with my PhD student Jun Chai. The formulation of the security notion follows our recent work with Sean Phillips, Alessandra Duz, and Fabio Pasqualetti. Part of the work presented here has recently appeared in conference venues and journal publications, and associated papers are available at <https://hybrid.soe.ucsc.edu>. I would also like to thank the support received to fund part of this work from the National Science Foundation under CAREER Grant no. ECS-1450484 and Grant no. CNS-1544396, the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, as well as CITRIS and the Banatao Institute at the University of California.

References

1. Y. Li, S. Phillips, and R. G. Sanfelice, "On distributed observers for linear time-invariant systems under intermittent information constraints," in *Proceedings of 10th IFAC Symposium on Nonlinear Control Systems*, 2016, pp. 654–659.
2. —, "Robust distributed estimation for linear systems under intermittent information," *To appear in IEEE Transactions on Automatic Control*, 2018.
3. S. Phillips, Y. Li, and R. G. Sanfelice, "On distributed intermittent consensus for first-order systems with robustness," in *Proceedings of 10th IFAC Symposium on Nonlinear Control Systems*, 2016, pp. 146–151.
4. S. Phillips, R. G. Sanfelice, and R. S. Erwin, "On the synchronization of two impulsive oscillators under communication constraints," in *Proceedings of the American Control Conference*, 2012, pp. 2443–2448. [Online]. Available: <https://hybrid.soe.ucsc.edu/files/preprints/61.pdf>
5. S. Phillips and R. G. Sanfelice, "A framework for modeling and analysis of robust stability for spiking neurons," in *Proceedings of the American Control Conference*, June 2014, pp. 1414–1419.
6. S. Phillips and R. Sanfelice, "Synchronization of two linear systems over intermittent communication networks with robustness," in *Proceedings of the IEEE Conference on Decision and Control*, December 2015, pp. 5569–5574.
7. S. Phillips and R. G. Sanfelice, "On asymptotic synchronization of interconnected hybrid systems with applications," in *Proceedings of the American Control Conference*, 2017, pp. 2291–2296.
8. S. Phillips, A. Duz, F. Pasqualetti, and R. G. Sanfelice, "Recurrent attacks in a class of cyber-physical systems: Hybrid-control framework for modeling and detection," in *To appear in Proceedings of the IEEE Conference on Decision and Control*, 2017.
9. R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, pp. 1645–1662, 1990.

10. S. Bohacek, J. P. Hespanha, J. Lee, and K. Obraczka, "Modeling communication networks using hybrid systems," *IEEE/ACM Trans. on Networking*, vol. 15, no. 6, pp. 663–672, June 2007.
11. A. Suri, J. Baillieul, and D. V. Raghunathan, "Control using feedback over wireless ethernet and bluetooth," in *Handbook of networked and embedded control systems*. Springer, 2005, pp. 677–697.
12. R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. New Jersey: Princeton University Press, 2012.
13. Y. Li and R.G.Sanfelice, "Results on finite time stability for a class of hybrid systems," in *Proceedings of the American Control Conference*, 2016, pp. 4263–4268.
14. J. Chai and R. Sanfelice, "On notions and sufficient conditions for forward invariance of sets for hybrid dynamical systems," in *Proceedings of the 54th IEEE Conference on Decision and Control*, December 2015, pp. 2869–2874.
15. J. Chai and R. G. Sanfelice, "Results on feedback design for forward invariance of sets in hybrid dynamical systems," in *Proceedings of the 55th IEEE Conference on Decision and Control*, December 2016, pp. 622–627.
16. —, "On robust forward invariance of sets for hybrid dynamical systems," in *Proceedings of the American Control Conference*, 2017, pp. 1199–1204.
17. C. Cai and A. R. Teel, "Characterizations of input-to-state stability for hybrid systems," *Syst. & Cont. Letters*, vol. 58, pp. 47–53, 2009.
18. R. G. Sanfelice, "Robust asymptotic stabilization of hybrid systems using control lyapunov functions," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, April 2016, pp. 235–244.
19. R. Yuan, Z. Jing, and L. Chen, "Uniform asymptotic stability of hybrid dynamical systems with delay," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 344–348, Feb 2003.
20. X. Liu and J. Shen, "Stability theory of hybrid dynamical systems with time delay," *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 620–625, April 2006.
21. Y. Sun, A. N. Michel, and G. Zhai, "Stability of discontinuous retarded functional differential equations with applications," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1090–1105, Aug 2005.
22. P. Yan and H. Özbay, "Stability analysis of switched time delay systems," *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 936–949, 2008. [Online]. Available: <https://doi.org/10.1137/060668262>
23. W.-H. Chen and W. X. Zheng, "Input-to-state stability and integral input-to-state stability of nonlinear impulsive systems with delays," *Automatica*, vol. 45, no. 6, pp. 1481 – 1488, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109809000867>
24. J. Liu, X. Liu, and W.-C. Xie, "Input-to-state stability of impulsive and switching hybrid systems with time-delay," *Automatica*, vol. 47, no. 5, pp. 899 – 908, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109811000768>
25. X.-M. Sun and W. Wang, "Integral input-to-state stability for hybrid delayed systems with unstable continuous dynamics," *Automatica*, vol. 48, no. 9, pp. 2359 – 2364, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109812003226>
26. A. Banos and A. Barreiro, "Delay-independent stability of reset systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 341–346, Feb 2009.
27. A. Barreiro and A. Baos, "Delay-dependent stability of reset systems," *Automatica*, vol. 46, no. 1, pp. 216 – 221, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109809004907>
28. J. Liu and A. R. Teel, "Lyapunov-based sufficient conditions for stability of hybrid systems with memory," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1057–1062, April 2016.

29. —, *Hybrid Dynamical Systems with Finite Memory*. Springer International Publishing, 2016, pp. 261–273. [Online]. Available: https://doi.org/10.1007/978-3-319-18072-4_13
30. —, “Invariance principles for hybrid systems with memory,” *Nonlinear Analysis: Hybrid Systems*, vol. 21, pp. 130 – 138, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1751570X15000515>
31. K.-Z. Liu and X.-M. Sun, “Razumikhin-type theorems for hybrid system with memory,” *Automatica*, vol. 71, pp. 72 – 77, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109816301637>
32. B. Altm and R. G. Sanfelice, “On Robustness of Pre-Asymptotic Stability to Delayed Jumps in Hybrid Systems,” 2017, submitted.
33. S. Phillips and R. G. Sanfelice, “Robust synchronization of interconnected linear systems over intermittent communication networks,” in *Proceedings of the American Control Conference*, 2016, pp. 5575–5580.
34. Y. Li and R. G. Sanfelice, “Interconnected observers for robust decentralized estimation with performance guarantees and optimized connectivity graph,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 1–11, March 2016.
35. S. Wang and W. Ren, “On the consistency and confidence of distributed dynamic state estimation in wireless sensor networks,” in *Proc. 54th IEEE Conference on Decision and Control*, Osaka, Japan, December 2015, pp. 3069–3074.
36. A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Information weighted consensus filters and their application in distributed camera networks,” *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, Dec 2013.
37. S. Park and N. C. Martins, “Design of distributed LTI observers for state omniscience,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1–16, 2017.
38. W. Zhang, L. Yu, and G. Feng, “Optimal linear estimation for networked systems with communication constraints,” *Automatica*, vol. 47, no. 9, pp. 1992 – 2000, 2011.
39. P. Naghshtabrizi and J. P. Hespanha, “Designing an observer-based controller for a network control system,” in *Proc. of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 848–853.
40. D. Dačić and D. Nešić, “Observer design for wired linear networked control systems using matrix inequalities,” *Automatica*, vol. 44, no. 11, pp. 2840 – 2848, 2008.
41. F. Dörfler, F. Pasqualetti, and F. Bullo, “Continuous-time distributed observers with discrete communication,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 296–304, 2013.
42. R. Postoyan and D. Nešić, “A Framework for the Observer Design for Networked Control Systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1309–1314, 2012.
43. T. Ahmed-Ali, I. Karafyllis, and F. Lamnabhi-Lagarrigue, “Global exponential sampled-data observers for nonlinear systems with delayed measurements,” *Systems & Control Letters*, vol. 62, no. 7, pp. 539 – 549, 2013.
44. M. Farza, M. M’Saad, M. L. Fall, E. Pigeon, O. Gehan, and K. Busawon, “Continuous-discrete time observers for a class of mimo nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 1060–1065, April 2014.
45. T. N. Dinh, V. Andrieu, M. Nadri, and U. Serres, “Continuous-discrete time observer design for Lipschitz systems with sampled measurements,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 787–792, March 2015.
46. F. Mazenc, V. Andrieu, and M. Malisoff, *Continuous-Discrete Observers for Time-Varying Nonlinear Systems: A Tutorial on Recent Results*, 2015, ch. 25, pp. 181–188.
47. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, “Foundations of control and estimation over lossy networks,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, Jan 2007.
48. L. Shi, L. Xie, and R. M. Murray, “Kalman filtering over a packet-delaying network: A probabilistic approach,” *Automatica*, vol. 45, no. 9, pp. 2134 – 2140, 2009.

49. R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conference on Decision and Control and 2005 European Control Conference*, Dec 2005, pp. 8179–8184.
50. F. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, Sept 2010.
51. J. Cortes, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, 2009.
52. F. Ferrante, F. Gouaisbaut, R. G. Sanfelice, and S. Tarbouriech, "State estimation of linear systems in the presence of sporadic measurements," *Automatica*, vol. 73, pp. 101–109, November 2016.
53. R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. New Jersey: Princeton University Press, 2012.
54. J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.
55. R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept 2004.
56. —, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3. IEEE, 2002, pp. 2965–2971.
57. H. Yamaguchi, T. Arai, and G. Beni, "A distributed control scheme for multiple robotic vehicles to make group formations," *Robotics and Autonomous systems*, vol. 36, no. 4, pp. 125–147, 2001.
58. J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.
59. R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the Conference on Decision and Control and European Control Conference*. IEEE, 2005, pp. 6698–6703.
60. A. Sarlette, R. Sepulchre, and N. Leonard, "Cooperative attitude synchronization in satellite swarms: a consensus approach," in *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace*, 2007, pp. 223–228.
61. N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.
62. J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726 – 737, 2008.
63. N. Jie and L. Zhong, "Sampled-data consensus for high-order multiagent systems under fixed and randomly switching topology," *Discrete Dynamics in Nature and Society*, vol. 2014, 2014.
64. H. Liu, G. Xie, and L. Wang, "Necessary and sufficient conditions for solving consensus problems of double-integrator dynamics via sampled control," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 15, pp. 1706–1722, 2010.
65. Z. Guan, Z. Liu, G. Feng, and M. Jian, "Impulsive consensus algorithms for second-order multi-agent networks with sampled information," *Automatica*, vol. 48, no. 7, pp. 1397–1404, 2012.
66. H. Hu, L. Yu, W. Zhang, and H. Song, "Group consensus in multi-agent systems with hybrid protocol," *Journal of the Franklin Institute*, vol. 350, no. 3, pp. 575–597, 2013.
67. G. Wen, Z. Duan, W. Yu, and G. Chen, "Consensus of multi-agent systems with nonlinear dynamics and sampled-data information: a delayed-input approach," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 6, pp. 602–619, 2013.
68. R. Goebel, "Basic results on pointwise asymptotic stability and set-valued Lyapunov functions," in *49th IEEE Conference on Decision and Control*. IEEE, 2010, pp. 1571–1574.

69. S. P. Bhat and D. S. Bernstein, "Nontangency-based lyapunov tests for convergence and stability in systems having a continuum of equilibria," *SIAM Journal on Control and Optimization*, vol. 42, no. 5, pp. 1745–1775, 2003.
70. V. N. Murthy and E. E. Fetz, "Synchronization of neurons during local field potential oscillations in sensorimotor cortex of awake monkeys," *Journal of Neurophysiology*, vol. 76, no. 6, pp. 3968–3982, 1996.
71. I. Belykh, V. Belykh, and M. Hasler, "Generalized connection graph method for synchronization in asymmetrical networks," *Physica D: Nonlinear Phenomena*, vol. 224, no. 1–2, pp. 42 – 51, 2006.
72. Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-time semistability theory with applications to consensus protocols in dynamical networks," in *Proceedings of the American Control Conference*, July 2007, pp. 2411–2416.
73. L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557 – 2562, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109809003525>
74. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.
75. J. J. Slotine, W. Wang, and K. El-Rifai, "Contraction analysis of synchronization in networks of nonlinearly coupled oscillators," in *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, 2004.
76. D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.
77. S. Cai, P. Zhou, and Z. Liu, "Synchronization analysis of hybrid-coupled delayed dynamical networks with impulsive effects: A unified synchronization criterion," *Journal of the Franklin Institute*, vol. 352, no. 5, pp. 2065 – 2089, 2015.
78. J. Lu, D. W. C. Ho, and J. Cao, "A unified synchronization criterion for impulsive dynamical networks," *Automatica*, vol. 46, no. 7, pp. 1215 – 1221, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109810001627>
79. T. Liu, M. Cao, C. D. Persis, and J. M. Hendrickx, "Distributed event-triggered control for synchronization of dynamical networks with estimators," *IFAC Proceedings Volumes*, vol. 46, no. 27, pp. 116 – 121, 2013.
80. C. D. Persis and P. Frasca, "Self-triggered coordination with ternary controllers," *IFAC Proceedings Volumes*, vol. 45, no. 26, pp. 43 – 48, 2012.
81. T. Liu, D. J. Hill, and B. Liu, "Synchronization of dynamical networks with distributed event-based communication," in *2012 IEEE 51st IEEE Conference on Decision and Control*, Dec 2012, pp. 7199–7204.