

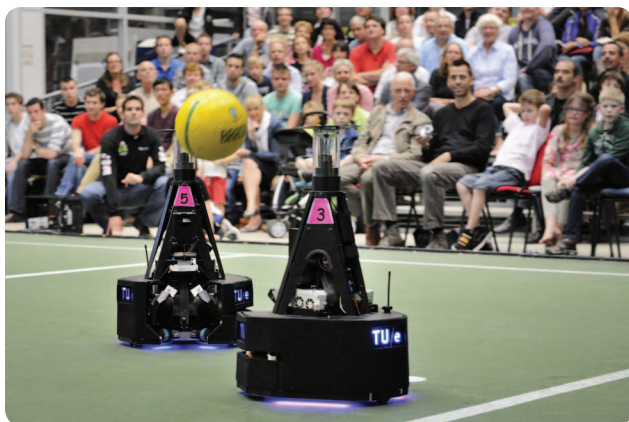
# Exploring the Boundaries of Robust Stability Under Uncertain Communication

## AN NCS TOOLBOX APPLIED TO A WIRELESS CONTROL SETUP

NICOLAS W. BAUER,  
S.J.L.M. (BAS) VAN LOON,  
NATHAN VAN DE WOUW, and  
W.P.M.H. (MAURICE) HEEMELS

**T**he past three decades have seen an explosive growth of the body of theoretical research in the area of networked control systems (NCSs). The immense interest for this subject is justified by the fact that society can benefit enormously if the control community can break free from the dependence on dedicated point-to-point wiring for communication between sensors, actuators, and controllers. Societal

Digital Object Identifier 10.1109/MCS.2014.2320394  
Date of publication: 14 July 2014



**FIGURE 1** A team of robots autonomously playing a game of soccer in the Robocup 2013 Championship. Wireless communication is used to share information between the robots as well as coordinate offensive and defensive strategies. (Image courtesy of Tech United Eindhoven.)

benefits can stem from several networked control application areas, which are currently being explored by researchers and engineers around the world.

## NETWORKED CONTROL APPLICATION AREAS

To show the relevance of networked control, a few specific applications that are related to problems of high societal impact are highlighted.

One such application area is concerned with the cooperative control of multiple agents, such as vehicles and robots, which operate on land, air, and/or sea. The applications in this area range from autonomously achieving a safe formation of unmanned aerial vehicles [19] to autonomously executing a team-based strategy of robots playing soccer [59] (see Figure 1) to autonomously operating a large (high-traffic) warehouse [30]. One specific application within this range is cooperative adaptive cruise control (CACC). The work on CACC studies the benefits of having vehicles communicate wirelessly in an effort to reduce traffic congestion and fuel consumption by cooperatively regulating the spacing between vehicles that are following each other [1], [52], [53], [55], [62]; see Figure 2. In fact, wireless



**FIGURE 2** A platoon of vehicles that cooperatively regulates the intervehicle spacing to improve traffic flow. Wireless communication is used to transmit acceleration setpoints between adjacent vehicles. (Image courtesy of NWO, The Netherlands.)

communication between vehicles is the key enabling technology for the implementation of CACC. As a consequence, the uncertainty inherent in wireless communication is unavoidable. This network-induced uncertainty forms an interesting challenge for such a safety-critical, real-time control application since it may produce unsatisfactory performance or could even cause vehicles to collide if not dealt with appropriately.

Another application area, where wireless communication must be used for real-time feedback, is formed by certain classes of renewable energy systems. Kite power systems are part of a family known as airborne wind energy (AWE) systems and, like wind turbines, are used for converting wind energy to electrical energy. However, AWE systems offer numerous advantages over the current technique of using massive turbine structures such as 1) lower environmental impact, 2) drastic reductions in material and installation costs, and 3) the potential to capture more wind energy due to operation at higher altitudes. A promising class of kite systems generates its electricity by pulling a tether that is anchored at a base-station [25], [37]. Communication between the kite and the base station is required because, to generate power, the base station must reel the tether in or out based on the kite's position and/or velocity. The distance between the kite and base station and the strong forces that are exerted on the tether makes wired communication between the kite and base-station undesirable—if not impossible. As such, a wireless control system (WCS) must be properly designed to maximize the energy-harvesting efficiency of this system.

Another application area of real-time control based on a shared communication network, that is, where sensor, controller, and actuator nodes share a common communication medium/bus, is the operation of large-scale systems such as electrical power distribution networks [10], water transportation networks [12], industrial factories [48], and energy collection networks, such as wind farms [38]. Installing dedicated wires across even hundreds of meters for a particular application becomes expensive and reduces the flexibility of the network layout. Using existing telecommunication networks offer a technological solution for the otherwise expensive implementation (of the control strategies) based on installing dedicated wiring between all control devices.

Given the specific application domains mentioned above, being able to analyze and design NCSs, thereby guaranteeing their reliability, may lead to a significant societal impact including the benefits of relieving traffic congestion on highways, cheaper and more environmentally friendly energy-harvesting systems, and efficient regulation of city-wide systems that are vital to modern society. To obtain these benefits, the theory developed by the NCS community must be delivered in the form of user-friendly design/analysis tools to the control engineers who ultimately design and implement control strategies and

define network requirements to meet certain closed-loop stability, performance, and reliability specifications.

## NETWORKED CONTROL THEORY AND VALIDATION CHALLENGES

The NCS community has provided a multitude of different valuable theoretical techniques to determine, to different extents, which properties (such as stability and performance) a particular NCS configuration possesses; see the surveys [8], [36], and [72]. Analyzing properties of an NCS that considers only linear plants and controllers is already a challenging problem since inserting a shared network in between sensors, actuators, and controllers induces uncertain data-rate variations, which result in nonlinear and hybrid (stochastic) phenomena in the control loop. Many researchers have addressed this challenge and contributed many different solutions to this difficult problem. Some researchers focused on the case where stochastic information is included in models for delays and transmission intervals [4], [20], [47], [60], [63]–[65]. Other researchers have focused on the “probability distribution free” case in [6], [15], [16], [21], [27], [28], [31], [49], [50], [68], and [72]. In the latter case, the time-varying and uncertain transmission intervals and transmission delays are taken from a bounded set, without presuming any knowledge on the particular probability distribution on this bounded support.

To bring these theoretical advances to real applications, such as mentioned in the previous section, they must be validated on practical setups. Before NCS theory can be validated experimentally, it (typically) must first be implemented in software to gain access to the analysis and design potential of such NCS theory and support model-based testing of experimental scenarios. However, the amount of time required to implement most techniques in software is large due to the complexity involved with the automated construction and checking (or solving) of a set of conditions that provide a solution to the analysis or design problem at hand. These complexities often include properly setting up nonlinear minimization problems, semidefinite programs (SDPs), and sometimes even bilinear matrix inequality conditions, and then evoking an appropriate solver. Moreover, the accuracy or effectiveness of the analysis when applied to a practical NCS is subject to uncertainty due to possible modeling limitations and/or conservativeness in the analysis and/or synthesis conditions. Hence, a considerable amount of time is required to implement (just) one of the many techniques available without any guarantee that the corresponding analysis is effective for a particular NCS.

Now is the time to investigate the practical applicability or effectiveness of the theoretical tools provided by the NCS community by

- » developing software that makes the analysis tools easily accessible by control engineers and easily applicable to a wide range of engineering problems

- » applying the software to different practical network settings and engineering systems to assess the effectiveness of different NCS analysis techniques.

The diverse application of such software would pave an experimentally driven path to identifying the most promising research directions and gears new (theoretical) advances accordingly. To help pioneer this research path, a prototype Matlab toolbox has been developed that contains new theoretical contributions on NCS analysis and design and has been applied to an experimental setup to evaluate the effectiveness of the implemented theoretical tools in practice.

## CRAFTING TOOLS FROM THEORY

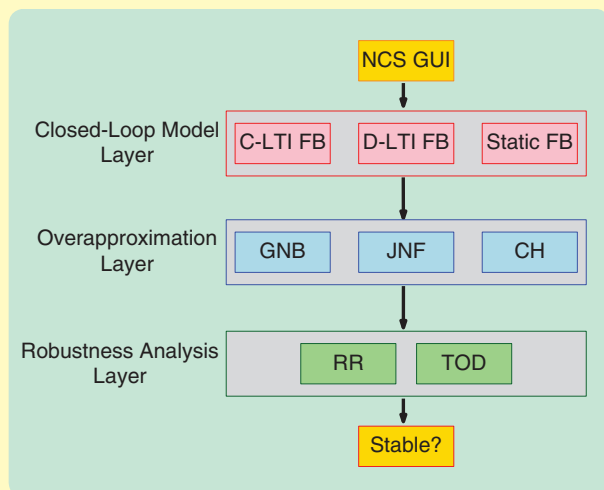
Control toolboxes are currently available that explicitly take into account time-varying network phenomena. Prominent examples include Truetime and Jitterbug [13], and the general-purpose simulator Matlab/Simulink including SIMEvents. Truetime [13] offers the ability to model and simulate a wide variety of real-time systems taking into account detailed NCS aspects, which is achieved by emulating a real-time operating system, thereby allowing the user to specify tasks and properties such as activation time, absolute deadline, and response time. The user can also indicate how tasks are scheduled, which leads to a complete simulation environment for NCSs. Jitterbug [13] operates on a more abstract system level and offers the ability to analyze performance in terms of a quadratic cost function for a Simulink model that switches between a finite set of stochastically varying delays. Truetime and Jitterbug provide the control engineer with necessary answers to questions related to designing control systems that operate reliably under uncertain communication. The prototype toolbox presented in this article provides the user with answers to different questions that are related to robust and stochastic stability analysis.

The Matlab toolbox described in this article has automated the theoretical developments in [4], [16], [21], [28], [31], and [44]. In [16], [21], [28], and [44], a discrete-time framework based on switched linear parameter-varying (SLPV) models was developed, and in [31] a continuous-time framework exploiting hybrid systems (as in [11], [29], and [51]) was developed to analyze the stability properties of NCSs. In the discrete-time framework, it has been shown that the amount of conservatism when analyzing stability can be made small; however, this framework is limited to the analysis of linear plants and (switched) linear controllers (with an exception being [68]). On the other hand, the (continuous-time) hybrid system formulation of the NCS dynamics has the advantage of explicitly including intersample behavior in the model and thus being able to more efficiently study the  $\mathcal{L}_p$ -gain or input-to-state stability-type performance criteria; see, for example, [31], [50], and [67]. As such, it is beneficial to consider both frameworks as each has its specific advantages. For the sake of brevity, this article focuses on the toolbox implementation of the discrete-time NCS analysis framework even though the analysis techniques in the hybrid system



## Software Structure and Customization

The overall structure of the Matlab robust stability analysis software, based on the discrete-time NCS framework [15], [16], [21], is given in Figure S1. The toolbox currently offers the



**FIGURE S1** A toolbox software structure for the discrete-time (switched) system framework. The software structure separates each of the three inner layers, which facilitates horizontal expansion of each layer individually.

functionality to 1) analyze three different controller structures, 2) employ three different overapproximation techniques (explained in “Robust Stability Analysis”), and 3) assess stability for two standard communication protocols. By carefully considering and implementing the software structure, horizontal expansion of each layer (for example, including different closed-loop models, different overapproximation techniques, and different analysis conditions) is straightforward since the vertical links are defined in a general manner (see [7] for more details).

The structure of the toolbox also reflects the objective of appealing to both basic and advanced users. The basic user can interact with the first and last layer of the toolbox through the GUI. For advanced users, the set of standard models provided with this toolbox might not include the exact model or stability analysis technique they are interested in studying, so the NCS toolbox allows users to customize one or more of the three levels shown in Figure S1 without affecting the others. This modular structure enables custom NCS configurations to be analyzed, custom overapproximations to be evaluated, and custom analysis conditions to be assessed. As a consequence, the toolbox offers the freedom to focus on a particular layer of interest and reduce the time needed to compare with existing techniques.

framework have been implemented for (linear) plants and controllers as well. An overview of the software structure for robust stability analysis using the discrete-time framework is shown in “Software Structure and Customization.”

By using this toolbox, the user can make multidisciplinary design tradeoffs between control properties (such as stability) and network-related properties (such as the bounds on transmission delays and intervals and the scheduling protocol employed) in a user-friendly manner. Currently, these multidisciplinary design tradeoffs can be made by means of 1) simulation capabilities and 2) the analysis results provided by the underlying theory, referenced above. In addition to implementing the theoretical results available in the literature, the toolbox also offers the possibility to identify basic control-relevant “communication models” (such as obtaining a probability distribution of the network delays and/or transmission intervals) based on measured data. Finally, it must be emphasized that there are many other results in the NCS literature that either use the discrete-time SLPV or the continuous-time hybrid system framework, which are currently not implemented in the toolbox. The hope is that the prototype toolbox presented here serves as a basis to which the whole NCS community can contribute so more theoretical tools become available for easy use. Such future contributions may also focus on the aspect of controller *synthesis* for NCSs, which is currently not addressed in the toolbox as, in the authors’ opinion, theoretical developments on synthesis techniques

need further maturing before it is meaningful to pursue software implementation in this toolbox.

To illustrate the theoretical tools, the toolbox is applied to an experimental inverted pendulum setup involving a wireless sensor network (WSN). The controller computes control commands based on the received measurements from a WSN consisting of two sensor nodes. By logging the network data during the closed-loop experiments, it is observed that this fairly simple NCS exhibits very rich behavior in terms of the network-induced effects. The section “Network Characterization” shows that the mechanisms that produce this rich behavior are a result of the reliable radio communication (in terms of being able to drastically reduce the amount of packet dropouts) hardware and the standards to which the hardware adheres. To begin explaining these mechanisms, the operation of WSNs and WCSs is described next in some detail.

## WIRELESS CONTROL SYSTEMS

The design of cyberphysical systems (CPSs) [3], [41], [42], [66] in general, and WCSs in particular, requires the simultaneous consideration of communication, computation, and control aspects, in addition to physical phenomena. When wireless communication is used, particular attention has to be given to the control-critical resource of communication, as it is both limited and shared amongst other devices using the same carrier frequency. The sharing of a network and also certain propagation effects in the wireless medium, such as

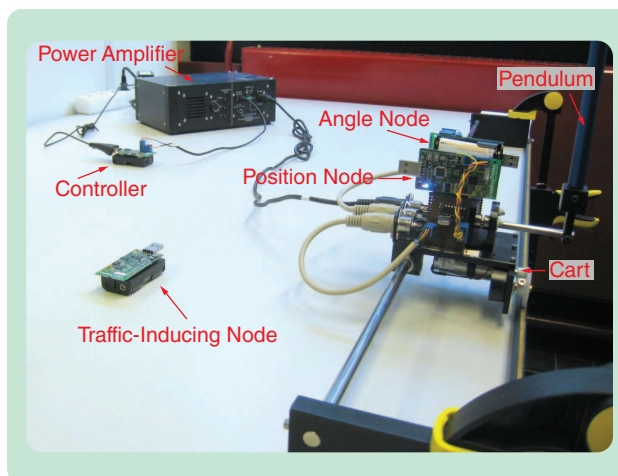
fading, induces uncertain data-rate variations for clients on that network. These data-rate variations result in nonlinear (hybrid) and possibly stochastically driven disturbances, which render the analysis and design of WCSs challenging.

A WCS typically uses a WSN to close the feedback loop. WSNs are allowed to operate in the unlicensed 2.4-GHz band, in which there are currently three major standards available: IEEE 802.11 for wireless local area networks (WLAN/Wi-Fi), IEEE 802.15.1 for wireless personal area networks (WPAN/Bluetooth), and IEEE 802.15.4 for low-rate wireless personal area networks (LR-WPAN). Due to its low power consumption and low cost, IEEE 802.15.4 is a very suitable candidate for battery-powered WSNs. The IEEE 802.15.4 standard specifies the physical layer and the media access control (MAC) layer for LR-WPANs. The ZigBee, ISA100.11a, WirelessHART, and MiWi specifications are different architectures, commonly referred to as “stacks,” all built upon the IEEE 802.15.4 standard in an effort to create a general networking architecture flexible enough to handle a variety of different applications. In fact, WSNs are currently being used in a variety of monitoring applications [14], [22], [39], [46], [69]. Despite the acceptance of using WSNs for monitoring applications, using wireless communication for real-time feedback control is still in its infancy. Monitoring applications do not have strict real-time requirements and do not suffer (as much) from data-rate or timing-related issues, compared to typical feedback control problems.

To overcome these timing issues, there are, roughly speaking, two approaches (in addition to codesign approaches for communication infrastructure and the control law, for which results are limited; see, for example, [71]). The first option is to optimize the communication infrastructure for a given controller, that is, to modify the radio communication stack or purchase the most expensive hardware available to attain as close to ideal communication as possible, which still may result in unsatisfactory behavior due to hardware limitations. This approach aims to improve the quality of service (QoS) of the network and was employed to achieve stabilization of an inverted pendulum/cart system using a similar wireless control experimental setup as used here [34]. The second option is to optimize the controller for a given communication infrastructure, that is, to develop and use (NCS) theory to be able to specify under what conditions slower and less expensive hardware can be used reliably, in the sense of still guaranteeing desirable closed-loop behavior. This article explores the second option. The focus of this article is on controller analysis and design using existing theoretical tools to preserve robustness of a given WCS, that is, a given WSN. In a more generic sense, the software tools provided here also support a multidisciplinary design approach for WCSs in which both network and control specifications are jointly optimized in an iterative fashion.

## THE WCS EXPERIMENTAL CASE STUDY

This article investigates the effects caused by network-induced disturbances when wirelessly controlling a pendu-

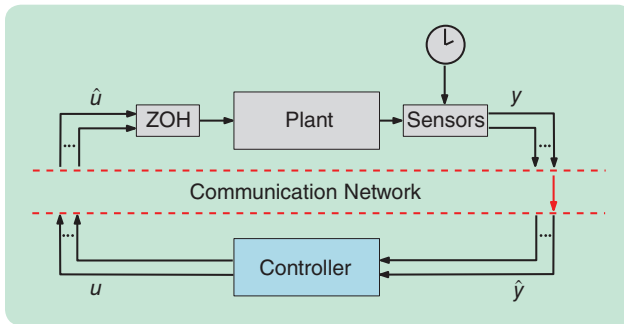


**FIGURE 3** A photograph of the wireless control experimental setup. The cart and pendulum are shown on the right. The position and angle transmitters are located on the cart, and they communicate their measurement data wirelessly to the controller (located on the left).

lum/cart system in an experimental setting; see Figure 3. When using wired communication, the pendulum/cart system is primarily used to investigate classical control theory since an accurate linear state-space model (valid within the desired range of operation) can be obtained from linearizing a first-principles nonlinear dynamic model [26]. A typical model of this setup consists of four states, two outputs, and a single control input. This setup requires output feedback controller synthesis techniques to design controllers with desired performance.

Several experimental results regarding WCSs can be found in the literature; see, for example, [23], [43], [56], and [70]. However, only a few wireless control experimental investigations jointly consider the control dynamics and time-varying network phenomena in the design and/or analysis; some examples include [23] and [56]. These works study different practical setups, such as the *rotary* inverted pendulum using Bluetooth [23] and Wi-Fi [56], and investigate different aspects than considered here.

In particular, the focus here is on the investigation of the effectiveness of newly developed robust and stochastic stability analysis techniques when used in combination with a controller design procedure, as well as on showcasing the features of the prototype NCS toolbox. A characterization of the communication network, in terms of the bounds on the transmission intervals and delays, is determined from experimental data. With these bounds in hand, developed theoretical tools [4], [7], [16], [21], [28], as implemented in the toolbox, can be applied and validated. The analysis performed using these tools provides both robust stability and stochastic stability regions for different performance specifications, which aid in *tuning* the controller to achieve both closed-loop stability properties with respect to the network-induced effects *and* meet performance specifications.



**FIGURE 4** A schematic of a networked control system. The plant and controller are separated by a communication network that induces uncertain data-rate variations, which result in nonlinear and hybrid (stochastic) phenomena in the control loop.

## MAIN CONTRIBUTIONS

The main contributions of this article are 1) the independent development of an experimental WCS setup, inspired by [34], and a prototype analysis toolbox, 2) the use of the NCS toolbox as part of a design process for a WCS including all the CPS aspects of physical behavior, control, computation, and communication, 3) the validation of theoretical NCS results (included in the NCS toolbox) in practice, and 4) the identification of challenges for the future of WCSs based on real-life experiments.

The article focuses on the presentation of a prototype NCS toolbox and its application to an experimental WCS setup. The relevant capabilities of the toolbox are highlighted in the form of sidebars throughout the article.

## NCS SETUP

The general setup of an NCS as considered in the NCS toolbox is depicted in Figure 4. The setup consists of a linear time-invariant (LTI) continuous-time plant and an LTI controller that are interconnected through a wired or wireless communication network, which induces the effects:

- » uncertain time-varying transmission intervals  $h$
- » uncertain time-varying network delays  $\tau$
- » uncertain dropout sequences
- » quantization
- » a shared communication medium, which prevents all sensor and actuator signals from being transmitted simultaneously.

Incorporating the communication network (including the protocol that arbitrates which sensor and actuator nodes are allowed to communicate at a transmission time) between the plant and controller leads to the specific operational aspects of the NCS. First, the sensor acts in a timedriven fashion, that is, tries to send data at each transmission time, and both the controller and the actuator act in an event-driven fashion, that is, respond instantaneously to newly arrived data. Second, the dropouts are modeled as prolongations of the transmission interval, meaning that, if a packet is considered “dropped,” then a new packet is transmitted at the next transmission time with new data. Retransmissions of packets can also be mod-

eled as prolongations of the delays by assuming that the node retransmits the same dropped message without resampling; see, for example, [15] and [61]. The number of consecutive packet dropouts is assumed to be bounded, and the probability distribution is unknown, although a stochastic analysis could be included as well; see, for example, [60], [62], and [65]. Third, the discrete-time control commands are converted to a continuous-time control signal by using a zero-order-hold (ZOH) function; see Figure 4. Fourth, a quantizer can be specified as fixed-step or logarithmic; see [18], [24], and references therein. Finally, the delays are assumed to be smaller than the transmission intervals. The aforementioned network-induced effects typically cause the plant output  $y$  and the control input  $u$  to not be equal to their networked versions  $\hat{y}$  and  $\hat{u}$ , respectively, as indicated in Figure 4. Such an NCS model can be conveniently created by using the NCS editor graphical interface; see “NCS Editor.” More information regarding the mathematical modeling and assumptions can be found in [15], [16], [21], [31], and [44].

It is generally known that any of the five aforementioned phenomena degrade performance and can even threaten closed-loop stability; see, for example, [16], [36], and [72]. Depending on the network and/or application, the influence or importance of each of these five effects can vary significantly. For example, systems that are able to transmit sensor data in one packet to the controller do not suffer from a shared communication medium but might take longer to collect all the data and prepare the transmission of the packet, resulting in delays (latencies) being dominant. Another example is the case where sensors share a wired controller area network (CAN) bus, thereby certainly being an application with a shared communication medium, but packet dropouts and significant delays rarely occur due to the high QoS of the CAN communication infrastructure. Hence, it is important to study the influence of different network-induced phenomena on closed-loop stability and performance. As a consequence, a general modeling and analysis framework is needed to deal with these different effects. Some applications exhibit all five effects simultaneously, as is the case for the final example described in this article. The setup for the wireless stabilization of this experimental pendulum/cart system is described in detail in the next section.

## DESCRIPTION OF THE WIRELESS CONTROL SETUP

The experimental setup consists of a pendulum/cart system and TelosB motes used for wireless transmission, as shown in Figure 3. The corresponding NCS, schematically depicted in Figure 5, consists of two sensor nodes, one node that measures the cart’s position and one node that measures the pendulum’s angle. Both sensor nodes wirelessly transmit their data separately to the controller, thereby resulting in a shared communication medium. The controller, however, is directly wired to the actuator. Each of the sensors and the controller are implemented on separate TelosB motes, as shown in Figure 3. Next, the pendulum/cart system is

described, the TelosB motes are discussed, and the communication logic that the motes obey is specified.

### Pendulum/Cart System

The physical setup is a freely swinging pendulum attached to a cart, shown schematically in Figure 6. This system has two degrees of freedom, resulting in a model with four states. The cart-pendulum setup has two optical encoders that measure the position of the cart and the angle of the pendulum. The

actuator is a dc motor that is located on the cart, which can drive the cart along one translational degree of freedom. The first-principles-based dynamic model of this system (also used in [34]) was derived in [58]. The model linearized around the upright open-loop position of the pendulum is

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx,\end{aligned}\quad (1)$$

where  $x = [x_c \ \dot{x}_c \ \theta \ \dot{\theta}]^T$ ,  $x_c$  is the position of the cart (m),  $\dot{x}_c$  is the cart velocity (m/s),  $\theta$  is the angle of the pendulum (rad),  $\dot{\theta}$  is the angular velocity of the pendulum (rad/s), and  $u$  is the input force (N) applied by the dc motor. The matrices in (1) are

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2.2755 & -6.8848 & -0.0089 \\ 0 & 29.0261 & -15.9037 & -0.0935 \end{bmatrix},$$

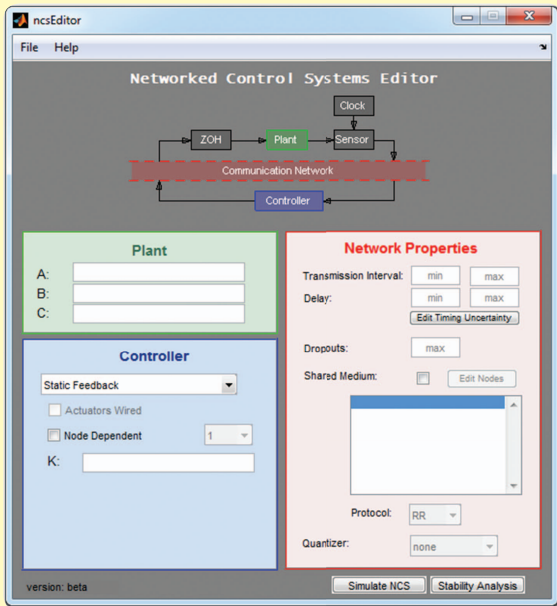
$$B = \begin{bmatrix} 0 \\ 0 \\ 1.6011 \\ 3.6985 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

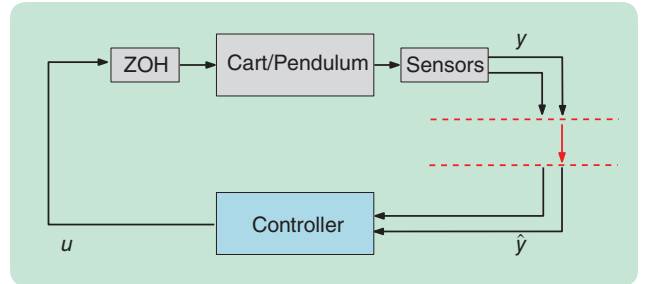
### NCS Editor

In the toolbox, NCS class objects describe models such as those discussed in the section “NCS Setup.” These NCS objects consist of a linear time-invariant (LTI) plant, an LTI controller, and network variables (that is, bounds on time-varying transmission intervals, bounds on time-varying delays, a bound on the maximum number of successive dropouts, the type of quantizer, the network node definitions, and the communication protocol). Creating an NCS object is easily done by using the NCS editor GUI, shown in Figure S2, since the variables are displayed and can be edited in a very intuitive manner.

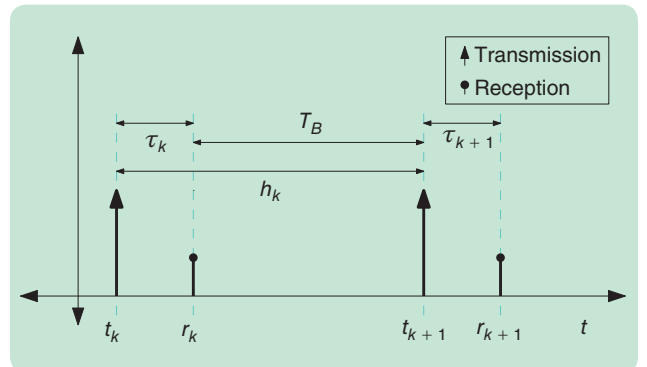
From this interface, links to the simulation functionality (see “Simulation”) and robust stability analysis functionality (see “Robust Stability Analysis”) are provided, which enables the user to modify the properties of the plant, controller, or network, and then run a simulation or assess robust stability to observe the consequences.



**FIGURE S2** A graphical user interface designed to define a networked control system (NCS) object. Similar to the NCS block diagram shown in Figure 4, the interface is divided into three sections: plant, controller, and network properties. Once data are input into these three sections, robust stability analysis and simulation can be directly performed.



**FIGURE 5** A block schematic of the experimental wireless control setup. The two sensor measurements are communicated separately over a shared wireless link subject to uncertain data-rate variations. The actuator link is considered to be wired.



**FIGURE 6** A mechanical schematic of the pendulum/cart system. The system consists of a freely swinging pendulum attached to a cart. The angle of the pendulum is denoted  $\theta$ , and the position of the cart is denoted  $x_c$ . The force input  $u$  can drive the cart along one translational degree of freedom.



## Time-varying transmission intervals, time-varying delays, and a shared communication medium are inherently present in the WCS.

resulting in the eigenvalues of the matrix  $A$  being  $\{0, 4.9929, -8.1257, -3.8455\}$ , indicating that the equilibrium of this system is open-loop unstable. Note that (1) does not contain a direct-feedthrough term  $D$ . Since a majority of physical models have transfer functions that are strictly proper, the toolbox does not support a direct-feedthrough term in the plant model. However, a direct-feedthrough term from  $\hat{y}$  to  $u$ , indicated in Figure 4, is supported for certain controller models.

The design objective is to synthesize an output-based controller that robustly stabilizes the origin of this system given that the measurements  $y$ , consisting of  $x_c$  and  $\theta$ , are transmitted wirelessly, thereby introducing network-induced imperfections on the sensor information available to the controller.

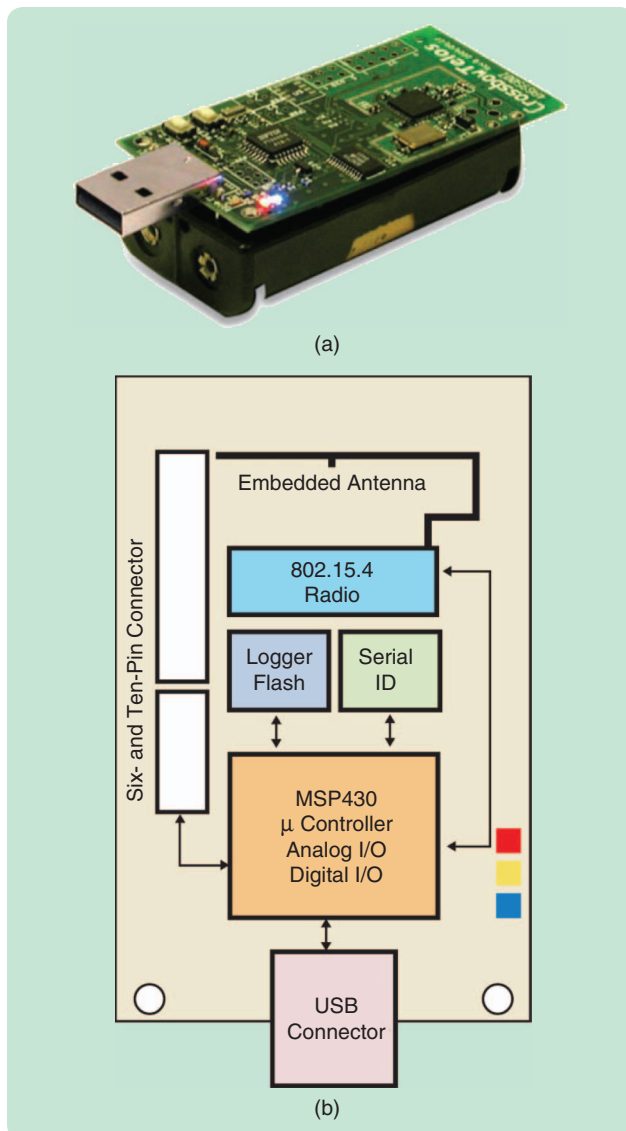
### TelosB Motes

The wireless devices that connect the two optical encoders to the controller are TelosB motes [57], shown in Figure 7. The TelosB motes are low-power embedded devices, which have been developed for quick prototyping purposes. Each TelosB mote communicates wirelessly by means of a CC2420 radio chip at a rate of 250 kb/s in the 2.4-GHz band and is IEEE 802.15.4 compliant. For computation, the devices have an 8-MHz TI MSP430 microcontroller with 48-kB ROM and 10-kB RAM. Finally, the devices are equipped with 1 MB of flash memory for data logging. The controller node uses the flash memory to store the received messages as well as controller information to be used for network and control signal analysis.

These devices are ideal for a wireless control experimental setup for several reasons. First, the devices are small, battery powered, wireless, and easily installed. Second, their development community is of considerable size, is still very active, and has forums that provide an invaluable resource for debugging. Third, the development environment for these embedded devices is very intuitive for users familiar with C programming, which results in quick and easy code adaptation. Finally, the computational capacity of the devices is limited, which poses an interesting theoretical challenge for designing a well-performing controller with low computational complexity.

### Communication Logic

For this experimental setup, a scheduling protocol must be implemented since each optical encoder reading is transmitted to the controller node separately. The round robin (RR) protocol, which operates by means of the channel access method known as time division multiple access, is implemented. The RR protocol requires that nodes communicate based on a fixed sequence that periodically repeats itself. This communication sequence prevents the radio transmission of the nearby nodes from interfering with each other. Since the nodes' clocks are not synchronized, a scheduling policy based on the nodes' internal clocks do not guarantee an RR-like operation. Two alternatives can solve this problem: 1) each node decides when to



**FIGURE 7** (a) A TelosB device and (b) schematic.



communicate based on when its predecessor node communicated or 2) there is a coordinator node that transmits evenly spaced beacons in time, indicating when the individual nodes are allowed to transmit.

The choice between these two communication strategies largely depends on the hardware used for wireless communication. Since TelosB devices are used, the wireless communication occurs by means of the CC2420 radio chip, which imposes the communication protocol to be IEEE 802.15.4 compliant. An IEEE 802.15.4 network has two modes of operation: a beacon-enabled mode or a nonbeacon mode. For this investigation, the standard TinyOS CC2420 radio stack (nonbeacon mode) is chosen as this communication strategy does not require any radio stack modification, whereas beacon-enabled communication does; see, for example, [17]. Each node decides when to transmit based on when it “hears” that the predecessor node in the fixed transmission sequence has transmitted. Once a node “hears” that its predecessor node communicated, the node first waits  $T_B$  seconds before it transmits to ensure that the controller has enough time to compute the previously sent control command and implement the control action before the next packet is received.

## CONTROLLER STRUCTURE AND CLOSED-LOOP MODELING

In this section, a more precise control objective is formulated and the structure of the controller(s), which is implemented on the TelosB devices, is defined. Then, a closed-loop model used for both robust and stochastic stability analysis is derived. Finally, to complete the model, the time-varying transmission intervals and time-varying delays are characterized based on network measurements from the experimental setup.

### Control Objective

To formulate the control objective, note that the model for which a controller is designed consists of the continuous-time plant model (1) and the network-induced effects (the communication model). The overall model is obtained by exactly discretizing the plant dynamics (1) leading to

$$\begin{aligned} x_{k+1} &= e^{A h_k} x_k + \int_0^{h_k} e^{A(h_k-s)} B u(t_k + s) ds \\ &= e^{A h_k} x_k + \int_0^{\tau_k} e^{A(h_k-s)} ds B u_{k-1} + \int_{\tau_k}^{h_k} e^{A(h_k-s)} ds B u_k \\ &= e^{A h_k} x_k + \int_{h_k-\tau_k}^{h_k} e^{A s} ds B u_{k-1} + \int_0^{\tau_k} e^{A s} ds B u_k \\ &= \hat{A}_{h_k} x_k + (\hat{E}_{h_k} B - \hat{E}_{h_k-\tau_k} B) u_{k-1} + \hat{E}_{h_k-\tau_k} B u_k, \\ \bar{y}_k &= C_{\sigma_k} x_{k_r}, \end{aligned} \quad (2)$$

where  $k \in \mathbb{N}$  is a counter related to the number of transmissions,  $h_k$  is the  $k$ th transmission interval (that is, the time between two transmissions),  $\tau_k$  is the  $k$ th delay, and  $x_k$  denotes the state of the plant at transmission time

$t_k := \sum_{\ell=0}^{k-1} h_\ell$  (where  $t_0 = 0$ ), that is,  $x_k = x(t_k)$ ,  $k \in \mathbb{N}$ . Similarly,  $y_k = y(t_k)$ ,  $k \in \mathbb{N}$ . In addition, the matrices  $\hat{A}_\rho := e^{A\rho}$ ,  $\hat{E}_\rho := \int_0^\rho e^{A s} ds$ , for  $\rho \in \mathbb{R}$ ,  $C_1 = [1 \ 0 \ 0 \ 0]$ , and  $C_2 = [0 \ 1 \ 0 \ 0]$ . The combinations  $(h_k, \tau_k)$  are presumed to be bounded inside a convex polygon  $\Omega$ , that is,  $(h_k, \tau_k) \in \Omega$  for all  $k \in \mathbb{N}$ , and  $\tau_k \leq h_k$  for all  $k \in \mathbb{N}$ , (small delay case). Finally, the packet reception times  $r_k := t_k + \tau_k$ , for all  $k \in \mathbb{N}$ . Figure 8 is provided to help clarify the relationships between the timing parameters  $h_k$ ,  $\tau_k$ ,  $t_k$ , and  $r_k$ .

The node that receives network access at transmission time  $t_k$ ,  $k \in \mathbb{N}$ , is denoted by  $\sigma_k$ , which due to the RR protocol (mentioned in the section “Communication Logic”), leads to

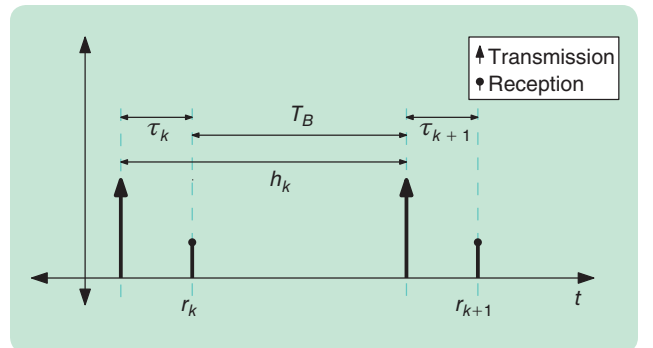
$$\sigma_k = \begin{cases} 1, & \text{for } k \text{ even,} \\ 2, & \text{for } k \text{ odd.} \end{cases} \quad (3)$$

Although the plant model (1) has two outputs, that is,  $y \in \mathbb{R}^2$ ,  $\bar{y}_k \in \mathbb{R}$  is used in (2) to explicitly model the fact that the controller only receives one measurement from only one sensor node at each transmission time  $t_k$ ,  $k \in \mathbb{N}$ .

Based on the above specifications, the informal control objective stated at the end of the section “Pendulum/Cart System” can now be translated to the more precise formulation of designing controllers that robustly (or at least stochastically) stabilize system (2) for all  $(h_k, \tau_k) \in \Omega$  for all  $k \in \mathbb{N}$ , as well as match (as closely as possible) a specified reference behavior, which indicates desired performance (detailed in the section “Controller Design” based on a wired setup with neglectable network effects). Descriptions of the robust and stochastic stability properties are given in the section “Controller Tuning for Nonideal Communication.”

### Control Structure

This section describes two observer-based controller structures (for reasons of comparison) that are implemented on a TelosB device and are used for wireless control experimentation. The low power of the TelosB devices limit the hardware resources available for computing control commands. In this respect, discrete-time controllers are

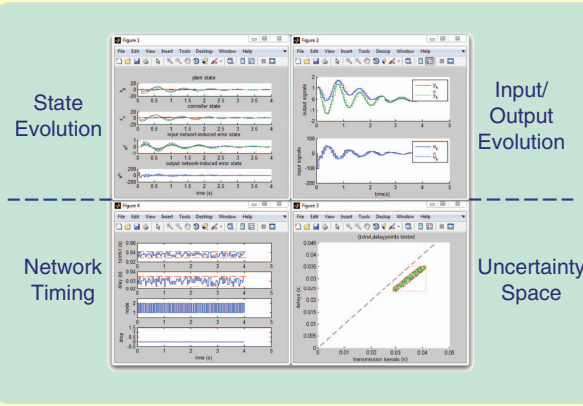


**FIGURE 8** A timing diagram that illustrates the relations between the node wait time  $T_B$ , the transmission times  $t_k$ , the transmission intervals  $h_k$ , the delays  $\tau_k$ , and the controller reception times  $r_k$ , where  $K \in \mathbb{N}$ .

## Simulation

To better familiarize a user with the behavior of the NCS models considered in the toolbox [such as in (7)], the user can numerically simulate the NCS once an NCS object is defined. To perform a simulation, the user can interact with a GUI that was designed to easily set initial conditions as well as to specify the sequences of transmission intervals, delays, and dropouts.

Once a simulation is finished, four plots that fully describe the state evolution of the NCS model are displayed. The top-left plot in Figure S3 contains state evolution information that



**FIGURE S3** The plots that result from performing a simulation of a networked control system object. The top two plots display control system performance data, and the bottom two plots display network-related data.

shows how the plant state  $x_k$  [described by (2)], the controller state  $\tilde{x}_k$  [described by (4) or (5)], and the network-induced errors  $e_k^y$  and  $e_k^u$  (see Remark 2) evolve. The top-right plot in Figure S3 contains input/output signal evolution information that shows how  $u_k$ ,  $\hat{u}_k$ ,  $y_k$ , and  $\hat{y}_k$  (see Figure S4) evolve, which provides quick insight into time-domain signal properties such as overshoot and settling time. The bottom-left plot in Figure S3 shows the evolution of the transmission intervals  $h_k$ , the delays  $\tau_k$ , the protocol sequence  $\sigma_k$ , and the dropout indicator  $\delta_k$ , (where  $\delta_k = 0$  and  $\delta_k = 1$  indicates whether a packet was received or dropped, respectively). Finally, the bottom-right plot in Figure S3 shows the  $(h, \tau)$  uncertainty space in red, which indicates the bounding polygon  $\Omega$  (defined in the section “Control Objective”) as well as the combinations of  $(h_k, \tau_k)$  used in the simulation in green. This plot visually conveys where a particular network resides in the uncertainty space and how a given uncertainty space is extended when dropouts occur. This  $(h, \tau)$  characterization shows that the assumption  $\tau_k \leq h_k$ ,  $k \in \mathbb{N}$ , in the section “Control Objective” is satisfied and is also used in the robustness analysis described in “Robust Stability Analysis.”

This simulation tool, combined with the ability to edit the network properties in the GUI (which includes the ability to include or remove network effects), enables the user to experiment and become familiar with the models that are used for robust stability analysis, and to gain insight in control performance-related consequences of network-induced imperfections.

preferred over continuous-time controllers for reasons of computational simplicity.

The first controller structure, which is implemented in a TelosB device, is in the form of a discrete-time observer-based controller, given by

$$\begin{aligned} \tilde{x}_{k+1} &= \bar{A}\tilde{x}_k + [\bar{B}_1 \ \bar{B}_2] \begin{bmatrix} u_k \\ u_{k-1} \end{bmatrix} + L(\hat{y}_k - C\tilde{x}_k), \\ u_k &= [K_x \ K_u] \begin{bmatrix} \tilde{x}_k \\ u_{k-1} \end{bmatrix}, \end{aligned} \quad (4)$$

and the second controller structure is a discrete-time *switched* observer-based controller given by

$$\begin{aligned} \tilde{x}_{k+1} &= \bar{A}\tilde{x}_k + [\bar{B}_1 \ \bar{B}_2] \begin{bmatrix} u_k \\ u_{k-1} \end{bmatrix} + L_{\sigma_k}(\bar{y}_k - C_{\sigma_k}\tilde{x}_k), \\ u_k &= [K_x \ K_u] \begin{bmatrix} \tilde{x}_k \\ u_{k-1} \end{bmatrix}, \end{aligned} \quad (5)$$

where  $\tilde{x}_k$  is the controller’s estimate of the plant state  $x_k$  and  $k \in \mathbb{N}$  again indicates the transmission number. As a result of the shared communication medium in (2),  $\hat{y}_k$  in (4) is specified as

$$\hat{y}_k = \Gamma_{\sigma_k}^y y_k + (I - \Gamma_{\sigma_k}^y) \hat{y}_{k-1}, \quad (6)$$

where  $\Gamma_1^y = \text{diag}(1, 0)$  and  $\Gamma_2^y = \text{diag}(0, 1)$ . Hence,  $\hat{y}_k$  can be interpreted as the most recent information the controller received of the plant output  $y$ . The matrices  $\bar{A} = \hat{A}_{h, \tau}$ ,  $\bar{B}_1 = \hat{E}_{h, \tau} B$ , and  $\bar{B}_2 = \hat{E}_{h, \tau} B - \hat{E}_{h, \tau - \tau} B$  for some  $h, > 0$ ,  $\tau, > 0$ , with  $\tau \leq h$ , indicate that (for computational simplicity) a nominal transmission interval  $h$ , and a nominal delay  $\tau$ , are used in the observer to embed a model of the plant (including nominal network parameters  $h$ , and  $\tau$ ). As already mentioned before, both controllers operate in an event-based fashion, in the sense that when they receive a new measurement packet, they immediately update their state and compute a new control action.

An important observation regarding the controllers is that the innovation term in (4) depends on  $\hat{y}_k \in \mathbb{R}^2$  whereas the corresponding term in (5) depends on  $\bar{y}_k \in \mathbb{R}$ . This difference means that, although both controllers only receive one measurement at each transmission time (due to network sharing), controller (4) heuristically corrects its state estimation with a “true” measurement and a delayed “pseudo” measurement, whereas (5) only uses the “true” measurement  $\bar{y}_k$ , as in (2). Hence, controller (5) can be designed to take into account the periodic switching caused by the RR protocol, whereas controller (4) cannot.

### Remark 1

The two controller structures (4) and (5), which were chosen for this specific experimental setup, are available for use in the prototype toolbox. The analysis of other commonly considered controller structures in the NCS literature and alternative network node configurations and protocols can also be modeled and analyzed using the toolbox (see “Software Structure and Customization”). More elaborate closed-loop configurations can also be included by appending the “Closed-Loop Model” layer, indicated in “Software Structure and Customization,” with additional NCS configurations.

### Closed-Loop Model

This section derives the closed-loop NCS model needed to analyze robust stability of an NCS that uses an observer-based controller as in (4). To create the closed-loop model, (2) and (4) are combined with (6), which results in the closed-loop system model

$$\bar{x}_{k+1} = \tilde{A}_{h_k, \tau_k, \sigma_k} \bar{x}_{k, \tau_k} \quad (7)$$

where  $\bar{x}_k = [x_k^\top \tilde{x}_k^\top u_{k-1}^\top \hat{y}_{k-1}^\top]^\top$  and  $\tilde{A}_{h_k, \tau_k, \sigma_k}$  is

$$\tilde{A}_{h_k, \tau_k, \sigma_k} = \begin{bmatrix} \hat{A}_{h_k} & \hat{E}_{h_k - \tau_k} B K_x & \hat{E}_{h_k} B - \hat{E}_{h_k - \tau_k} B (I - K_u) & 0 \\ L \Gamma_{\sigma_k}^y C & \bar{A} + \bar{B}_1 K_x - LC & \bar{B}_1 K_u + \bar{B}_2 & L(I - \Gamma_{\sigma_k}^y) \\ 0 & K_x & K_u & 0 \\ \Gamma_{\sigma_k}^y C & 0 & 0 & I - \Gamma_{\sigma_k}^y \end{bmatrix} \quad (8)$$

A closed-loop model for controller (5) can be constructed in a similar manner. The construction of such a closed-loop model is automatically performed in the toolbox based on the inputs specified through the NCS editor graphical user interface (GUI); see “NCS Editor.” Moreover, simulation of such closed-loop models is also conveniently included in the toolbox; see “Simulation.” Next, the control-relevant “communication model” is characterized in terms of bounds on the pairs of transmission intervals and delays  $(h_k, \tau_k)$ ,  $k \in \mathbb{N}$ , by analyzing network measurements that result from the communication logic described in the section “Communication Logic.”

### Remark 2

The closed-loop model (7) has  $u_{k-1}$  and  $\hat{y}_{k-1}$  as state variables. Commonly in the NCS literature, the network-induced errors  $e_k^u = \hat{u}_{k-1} - u_k$  and  $e_k^y = \hat{y}_{k-1} - y_k$  are used as states to describe the difference between what is the most recent information that is available at the controller/plant and the current value of the plant/controller output; see, for example, [15], [16], [21], [31], and [50]. Since there exists a similarity transformation between the models based on  $[\hat{u}_{k-1}^\top \hat{y}_{k-1}^\top]^\top$  and  $[e_k^u^\top e_k^y^\top]^\top$  (and the fact that the

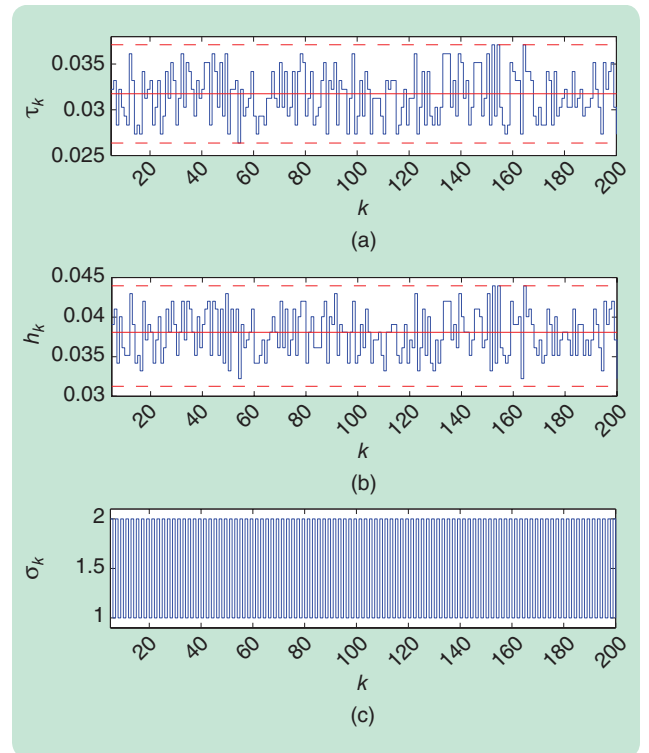
actuators are wired to the controller leads to  $u_{k-1} = \hat{u}_{k-1}$ ), the closed-loop model consisting of the states  $[\hat{u}_{k-1}^\top \hat{y}_{k-1}^\top]^\top$  is algebraically equivalent to the closed-loop model consisting of the states  $[e_k^u^\top e_k^y^\top]^\top$ . Since the two closed-loop models are algebraically equivalent, either state-space representation can be used to analyze the NCS with the RR protocol.

### Network Characterization

This section describes the network-induced effects that are present when using the TelosB motes for communication and control. The controller that is designed must be robust with respect to these (uncertain) effects.

#### Shared Communication

Since each of the sensor nodes is transmitting separately, the communication medium is shared. As explained in the section “Communication Logic,” the RR protocol has been implemented in the communication logic. From the  $\sigma_k$  plot of Figure 9, it can be confirmed that the communication logic described in the section “Communication Logic” results in the measurements being received by the controller in a RR fashion.



**FIGURE 9** Network measurements of delays  $\tau_k$ , transmission intervals  $h_k$ , and node access  $\sigma_k$ . The delay and transmission interval sequences are stochastic, whereas the node access sequence (dictated by the communication logic) is periodic. Hence, a combination of both stochastic and periodic switching phenomena are present in the wireless experimental setup.

## Delays

Throughout this article, delays  $\tau_k$ ,  $k \in \mathbb{N}$ , are the sum of the transmission delay and the computational delay. Measurements of the delays can be seen in the  $\tau_k$  plot of Figure 9. The transmission delay can be measured locally at each of the sensor nodes by programming the devices to transmit the time difference between when the `send` command is called and the `sendDone` event is signaled.

## Transmission Intervals

Measurements of the transmission intervals  $h_k$  are shown in the  $h_k$  plot of Figure 9. Unlike delays, transmission intervals (sampling) cannot be *directly* measured due to the facts that 1) the two sensor nodes decide when to transmit independently, 2) there is no clock synchronization between the devices, and 3) the controller only knows when messages are received, not transmitted. However, the transmission intervals can be reconstructed by using the time stamps of the controller's received sensor packets  $r_k$  and the transmission delay measurements  $\tau_k$ , which is done by noticing in Figure 8 that  $h_k + \tau_{k+1} = (r_{k+1} - r_k) + \tau_k$ .

As can be concluded, time-varying transmission intervals, time-varying delays, and a shared communication medium are *inherently* present in the WCS, that is, their influence is unavoidable given this network configuration. Although packet dropouts between the two sensor nodes are avoided due to the RR protocol, packet dropouts can still be caused by other wireless devices that are communicating in the surrounding environment. It has been experimentally verified by, for example, [45] and

[54] that the most dominant cause of packet dropouts is due to the coexistence of other devices that utilize the same (2.4 GHz) spectrum (such as Bluetooth, Wi-Fi, and other LR-WPANs). The collision avoidance CSMA/CA mechanism, used to ensure reliable radio communication, effectively avoids the possible interference caused by coexistence, that is, packet dropouts rarely occur. Interestingly, the CSMA/CA mechanism induces the variation of the transmission delays (and transmission intervals) to reduce the number of packet dropouts. Thereby, this mechanism increases the influence of one (or in this case two) network-induced effect(s) (transmission intervals and delays) to reduce that of another (packet dropouts).

The histogram, shown in Figure 10, characterizes the distribution of transmission intervals  $h$  and delays  $\tau$  shown in Figure 9. This  $(h, \tau)$ -space characterization 1) shows that the small-delay assumption  $\tau_k \leq h_k$  made in the section "Control Objective" is satisfied, 2) provides the distribution required for stochastic analysis, and 3) is also helpful for the robustness analysis in the section "Controller Tuning for Nonideal Communication." "Measurement-Based Network Characterization" briefly discusses the toolbox functionality regarding the  $(h, \tau)$  characterization.

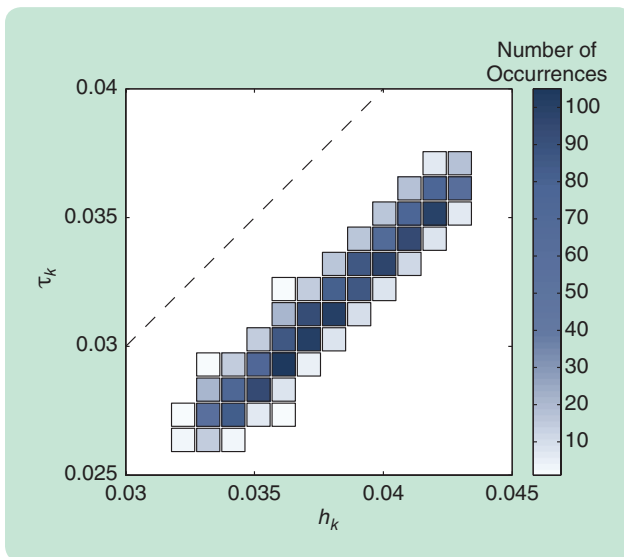
## CONTROLLER DESIGN

Now that the WCS setup has been formulated as a closed-loop system and the transmission intervals and delays have been characterized experimentally, controllers (4) and (5) are designed. Three controller types are compared, denoted by C0, C1, and C2:

- » **C0:** Controller (4) with  $h_* = 0.038$  and  $\tau_* = 0.000$
- » **C1:** Controller (4) with  $h_* = 0.038$  and  $\tau_* = 0.032$
- » **C2:** Controller (5) with  $h_* = 0.038$  and  $\tau_* = 0.032$ .

In words, controller C0 might be a typical controller setup ignoring any network-induced imperfections since it only includes nonzero nominal transmission interval information (and zero nominal delay) in its observer model. Controller C1 includes both nominal transmission interval and nominal delay information, and C2 includes both nominal transmission interval and delay information as well as taking periodic output switching due to the RR protocol into account. The values  $h_* = 0.038$  and  $\tau_* = 0.032$  were chosen as the average values of the measured transmission intervals and delays, respectively, as indicated in Figure 9. The *design* of C0 and C1 is based on the assumption that  $\hat{y}_k = y_k$  (that is, emulation-based design [11], [31], [40], [51]), meaning periodic output switching is not taken into account. However, the fact that  $\hat{y}_k \neq y_k$  but  $\hat{y}_k$  satisfies (6) is included explicitly in the robust stability analysis later.

To provide an indication of the performance that can be obtained with (near) perfect communication, that is, when all communication are wired, the "baseline" linear-quadratic-Gaussian (LQG) controller is synthesized in these ideal circumstances. The aim is to recover/approach this



**FIGURE 10** A histogram of measured transmission intervals  $h_k$  and delays  $\tau_k$ . This histogram shows that the small-delay assumption  $\tau_k \leq h_k$  is satisfied for the wireless experimental setup. The measurement data is clustered along the line  $h_k = \tau_k + T_B$  due to the specific communication logic chosen. Implementing a different communication logic is likely to result in a different  $(h, \tau)$  distribution.



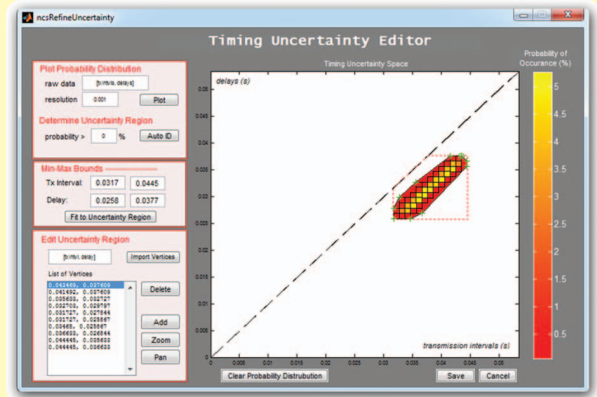
## Measurement-Based Network Characterization

Extracting nonconservative network characteristics from traffic data that are relevant for analysis can be, in itself, a tedious task. To alleviate the burden of this task, the software can plot and extract the relevant network properties needed for analysis on the basis of measured sequences of delay and transmission intervals.

In Figure S4, the graphical interface for measurement-based network characterization is shown. On the right, there is a plot of the  $(h, \tau)$ -uncertainty space, which is defined by delays  $\tau$  on the vertical axis and transmission intervals  $h$  on the horizontal axis. A histogram of actual measurement data is plotted by the software so that networks can be visualized and characterized in this space. On the left, there are options for inputting measurement data and options to automatically (and manually) define a region of interest. Automatically defining a region of interest allows the designer to immediately start simulating and analyzing after providing some network data to the toolbox. In the case where the automatic characterization is not providing a desired characterization, tools to edit the characterization manually are included.

The measurement-based functionality for network characterization also makes it possible to link the NCS toolbox with alternative toolboxes that simulate NCSs. Once a simulation

using, for example, Truetime [13], has been completed, the transmission interval and delay sequences can be imported into the network characterization tool, thereby enabling the NCS to be simulated and analyzed also with this toolbox.



**FIGURE S4** A networked control system timing-uncertainty editor graphical interface. An example network characterization is plotted on the right, and the control panel for editing the characterization is shown on the left. The characterization is done in the  $(h, \tau)$ -uncertainty space, which is defined by delays  $\tau$  on the vertical axis and transmission intervals  $h$  on the horizontal axis.

reference behavior using the wireless setup by retuning the observer-based controller (by selecting different weights in the typical LQG cost functions). Therefore, first, the baseline LQG controller is designed based on ideal communication circumstances, that is, using the controller C0 with  $\hat{y}_k = y_k$ ,  $h_k = h_*$ , and  $\tau_k = 0$ . The corresponding performance of C0 is experimentally evaluated in this “wired setting” so that it can be used for comparison to the situation in which sensor-to-controller communication is performed wirelessly. These weights are used as a basis to design C1 and C2 as well as propose a design procedure to tune (and, in fact, reduce) the controller aggressiveness to allow for more robustness to the network-induced effects using the NCS toolbox [7].

### Reference Behavior Under Ideal Communication

To design C0 for the wired setting, it is assumed that the separation principle holds [5], [35], that is, the optimal (Kalman) estimator and optimal LQR state-feedback controller can be designed separately.

The design of the optimal output injection gain,  $L$  in (4), is accomplished by solving the linear quadratic estimator (LQE) problem. A disturbance  $d_k \in \mathbb{R}^2$  and noise  $n_k \in \mathbb{R}^2$  are assumed to enter the plant dynamics (2) affinely, that is,

$$\begin{aligned} x_{k+1} &= \bar{A}x_k + [\bar{B}_1 \ \bar{B}_2] \begin{bmatrix} u_k \\ u_{k-1} \end{bmatrix} + Ed_k, \\ y_k &= Cx_k + n_k, \end{aligned} \quad (10)$$

and that the disturbance and the noise are uncorrelated zero-mean Gaussian stochastic processes with covariance matrices

$$\mathbb{E}[d_k d_k^T] = R_d, \quad \mathbb{E}[n_k n_k^T] = Q_d, \quad R_d, Q_d \succ 0, \quad (11)$$

where the matrices  $R_d$  and  $Q_d$  are weights chosen by the designer to achieve desired closed-loop behavior (possibly resembling the actual covariance matrices). For this setup, the main component in  $d$  is the disturbance that is entering the dc motor, so  $E = [\bar{B}_1 \ \bar{B}_2]$ . Since the controller (4) is in discrete time, the LQE problem is to find the matrix  $L$  in (4) that minimizes the *discrete-time* cost function  $\lim_{k \rightarrow \infty} \mathbb{E}[\|x_k - \tilde{x}_k\|^2]$ . Hence, the gain  $L$  was obtained by the standard discrete-time Kalman filter gain [2] based on the data  $\bar{A}$ ,  $\bar{B}_1$ ,  $\bar{B}_2$ ,  $C$ ,  $Q_d$ , and  $R_d$ . With some trial and error, choosing  $R_d = \text{diag}(10^5, 10^5)$ ,  $Q_d = I$  for C0 produces the baseline observer performance in terms of a desirable response, which is elaborated on later.

The design of the optimal state feedback gain  $K$  in (4) is accomplished by solving the linear quadratic regulator (LQR) problem, which is finding a control law  $u_k = Kx_k$  such that the cost function

$$J(x_0, \mathbf{u}) = \sum_{k=0}^{\infty} x_k^T Q x_k + \gamma^{-1} u_k^T R u_k \quad (12)$$

is minimized over  $\mathbf{u} = (u_0, u_1, \dots)$ . Similar to the LQE case, the matrices  $Q \succ 0$ ,  $R \succ 0$ , and the constant  $\gamma > 0$  are

## The article focuses on the presentation of a prototype NCS toolbox and its application to an experimental WCS setup.

weights chosen by the designer to achieve desired closed-loop behavior. The discrete-time plant model

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ I \end{bmatrix} u_k \quad (13)$$

was used to arrive at a solution to this problem. With some trial and error, choosing the matrices  $Q = \text{diag}(10^4, 10^4, 1, 1, 10^{-8})$ ,  $R = 1$ , and  $\gamma = 1$  for C0 results in a gain  $K$  producing the baseline controller performance in terms of a desirable response, which is elaborated on next. Note that the parameter  $\gamma$  can be increased or decreased to make the controller more aggressive or less aggressive, respectively.

Now that a controller C0 has been designed in the wired setting (that is, when  $h_k = 0.038$ ,  $\tau_k = 0$ , and  $\hat{y}_k = y_k$ ,  $k \in \mathbb{N}$ ), the meaning of baseline performance is now explained. Figure 11 shows the experimental data collected when  $\gamma = 1$ , using only wired communication. It can be concluded that the controller is able to keep the pendulum upright, however, there is an apparent steady-state oscillation on the cart's position, the pendulum's angle, and the

input signal. This oscillation is suspected to be primarily caused by friction-related nonlinearities in the translational support to the cart. Indeed, the cart does not respond to an input voltage that produces a force less than the opposing static friction force(s), and, as a consequence, the controller increases the force requested until this friction force is overcome, and the cart moves to “catch” the pendulum, thereby causing the steady-state oscillation. This phenomenon was also observed in experiments using a *rotary* inverted pendulum in [56]. An aggressive controller is desirable since it ramps up its input more rapidly and is able to keep the pendulum within a very small range. The response given by  $\gamma = 1$  is considered the baseline since it can stabilize the pendulum within acceptably narrow margins, while not producing too much noise on the dc motor input. As  $\gamma$  is decreased, the controller becomes less aggressive, and, consequentially, the set to which the pendulum can be stabilized by the controller increases, which is considered to be worse performance. Hence, the controller aggressiveness  $\gamma$  is used also as the performance characteristic as it can be directly related to the baseline performance, at least in the fully wired case.

To summarize, all the design parameters,  $h$ ,  $\tau$ ,  $Q$ ,  $R$ ,  $Q_d$ ,  $R_d$ , and  $\gamma$ , needed for designing the gains  $K$  and  $L$  for C0 have been specified. Taking  $\gamma = 1$  results in

$$K = [58.59 \quad -132.96 \quad 42.94 \quad -24.47 \quad 0.00],$$

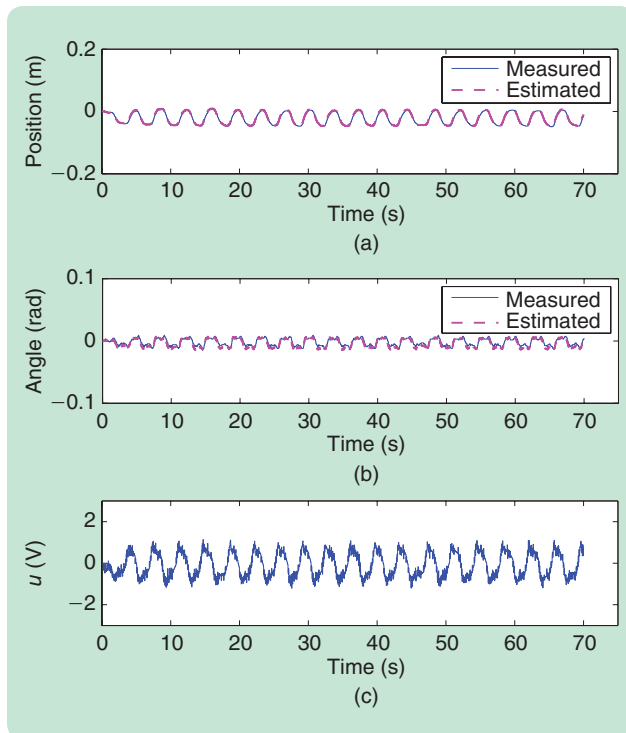
$$L = \begin{bmatrix} 0.37 & 0.42 & 2.26 & 4.65 \\ 0.42 & 1.20 & 4.47 & 11.93 \end{bmatrix}^T,$$

which produces the baseline reference performance for C0 in the *wired* control system. However, this choice might (and, in fact, does for C0) result in instability of the WCS due to the network-induced imperfections.

The next section determines values of  $\gamma$  that result in different control parameters that guarantee (different degrees of) robust stability for C0, C1, and C2. The controller C1 uses the same  $Q$ ,  $R$ ,  $Q_d$ , and  $R_d$  weights as C0. The switched controller C2 requires a pair  $(Q_{d_i}, R_{d_i})$ ,  $i \in \{1, 2\}$ , of weights for each of the two switching modes corresponding to  $\sigma_k \in \{1, 2\}$ , in (5). The matrices  $(Q_{d_i}, R_{d_i})$ ,  $i \in \{1, 2\}$ , are specified as  $R_{d_1} = R_{d_2} = R_d$ , and  $Q_d = \text{diag}(Q_{d_1}, Q_{d_2})$ . The same  $Q$  and  $R$  weights as for C0 and C1 are used. The gains  $K$ ,  $L_1$ , and  $L_2$  corresponding to C2 are designed using the *periodic* LQG technique described in [9], [33].

### Controller Tuning for Nonideal Communication

In this section, the closed-loop NCS model (7) is used along with the NCS toolbox so that the LQR controller parameter  $\gamma$



**FIGURE 11** Ideal baseline controller performance. Here, C0 is used with  $\gamma = 1$ , and sensor communication is wired. The aim is to recover/approach this reference behavior using wireless communication.

## Robust Stability Analysis

To put theoretical tools in the hands of control engineers, the toolbox completely automates the stability analysis procedure. The user is only required to input a plant model, a controller model, and (measurement) data regarding network uncertainties (described in the section “NCS Setup”) and then can directly start analyzing robust stability of the corresponding closed-loop NCS.

In the discrete-time framework, a polytopic overapproximation [32] of the closed-loop NCS model is necessary before LMIs can be used to determine if robust stability can be guaranteed. Such an overapproximation is needed because the  $h_k$  and  $\tau_k$  variables in (7) appear nonlinearly and take values within an infinite set. A polytopic overapproximation of, for example, (7), is a system that includes all possible dynamics of (7) and has a form suitable for deriving a finite set of LMI conditions, which, if satisfied, guarantee robust stability of (7). More formally, the system  $\bar{x}_{k+1} = A_k \bar{x}_k$  with  $A_k \in S_{\sigma_k}$ ,  $\sigma = 1, 2, \dots, N$ , is an overapproximation of (7) if  $\{\tilde{A}_{h,\tau,\sigma} \mid (h,\tau) \in \Omega\} \subseteq S_{\sigma}$  holds for all  $\sigma$ . This inclusion relation shows that any solution to (7) for a given protocol is also a solution to the overapproximation and the protocol. Moreover, an overapproximation is considered to be polytopic if, in addition to the above inclusion relation,  $S_{\sigma}$  has a polytopic structure, that is,  $S_{\sigma} = \{\sum_{i=1}^M \lambda_i A_{\sigma,i} + B_{\sigma} \Delta C_{\sigma} \mid \lambda_i \geq 0, \sum_{i=1}^M \lambda_i = 1, \Delta \in \Delta\}$  with  $A_{\sigma,i}, B_{\sigma}, C_{\sigma}$  given matrices for  $\sigma = 1, 2, \dots, N$ , and  $i = 1, 2, \dots, M$ , and  $\Delta$  is a structured set of matrices (often given by norm bounds). Acquiring a polytopic overapproximation (possibly with norm-bounded uncertainty) is the most tedious aspect of implementing a discrete-time approach for the stability analysis of NCSs. The toolbox automates the overapproximation procedure for a general class of models using a choice of different overapproximation techniques. Three overapproximation techniques are automated in this toolbox (as indicated in “Software Structure and Customization”). For an approach based on gridding and norm bounding (GNB), see, for example, [21], for an approach based on the Jordan normal form (JNF), see, for example, [16], and for an approach based on the Cayley-Hamilton theorem (CH), see, for example, [28]. A theoretical comparison between these three methods is given in [32].

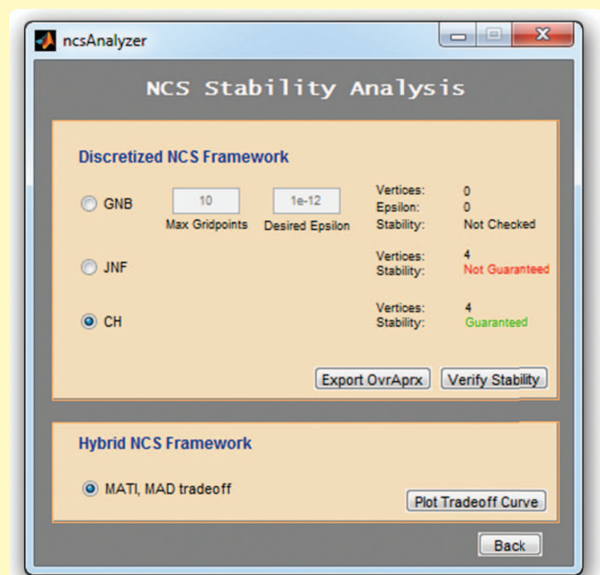
Within the analysis GUI, shown in Figure S5, it is possible to select one of the three automated overapproximation techniques (JNF, CH, or GNB). The overapproximated model can be generated by clicking on “Export OvrAprx” in the analysis GUI. Once the procedure has completed, a new variable is created in the workspace that contains the matrices corresponding to the overapproximated system as well as additional information related to the particular overapproximation chosen. For example, exporting a GNB overapproximation results in a variable being created where the additional information includes the Jordan blocks used for the overapproximation, the list of grid points used to generate the overapproximation, and the approximation error terms. More information on the GNB technique is provided in [21].

Although each of these three techniques is mathematically interesting, a fairly strong familiarity with the notation is required to implement (and actually use) these techniques in software.

Thus, for the control engineer who would like to determine if a specific closed-loop system is robust to network-induced effects, it costs a significant amount of time and effort to understand and implement these techniques. The NCS toolbox allows the control engineer to easily verify if a control setup possesses robustness properties without having to know all the (mathematical) details about a polytopic overapproximation besides the basic idea and the fact that some conservatism may be added.

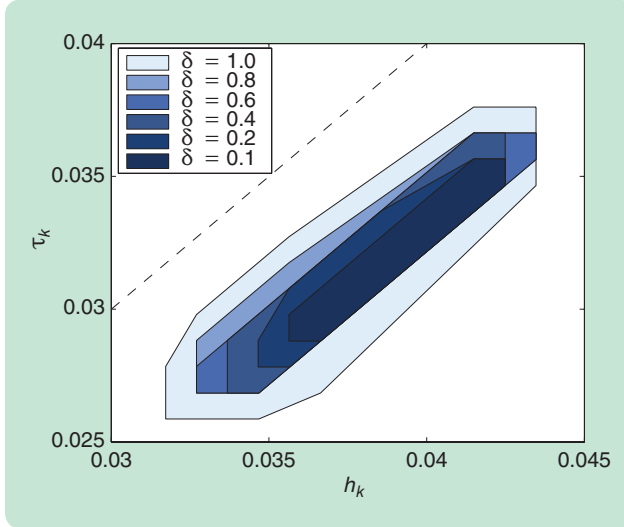
For example, the basic idea behind the GNB algorithm is to iteratively tighten an overapproximation by iteratively adding grid points to the overapproximated model at the location of the worst-case approximation error until either the user-specified maximum number of grid points is reached or the user-specified desired tightness of the overapproximation is achieved; see [21] and Figure S5. The resulting tightness of the overapproximation that is obtained, denoted  $\epsilon$ , is a norm related to the overapproximation error. This basic idea behind the GNB algorithm is all the user needs to know before starting to apply the technique in numerical examples.

Robust stability can be guaranteed if specific sets of LMIs that depend on the network-induced effects considered are satisfied. The toolbox automatically selects a suitable set of LMIs that correspond to the network-induced effects considered, which allows the user to perform stability verification within an analysis GUI (see Figure S5) so that the user can easily verify if stability of the NCS can be guaranteed. Combining the ability to directly verify stability with the ability to modify the network-induced effects, the user can experience firsthand which overapproximation techniques provide the least conservative results, and to what extent network-induced effects can be tolerated.

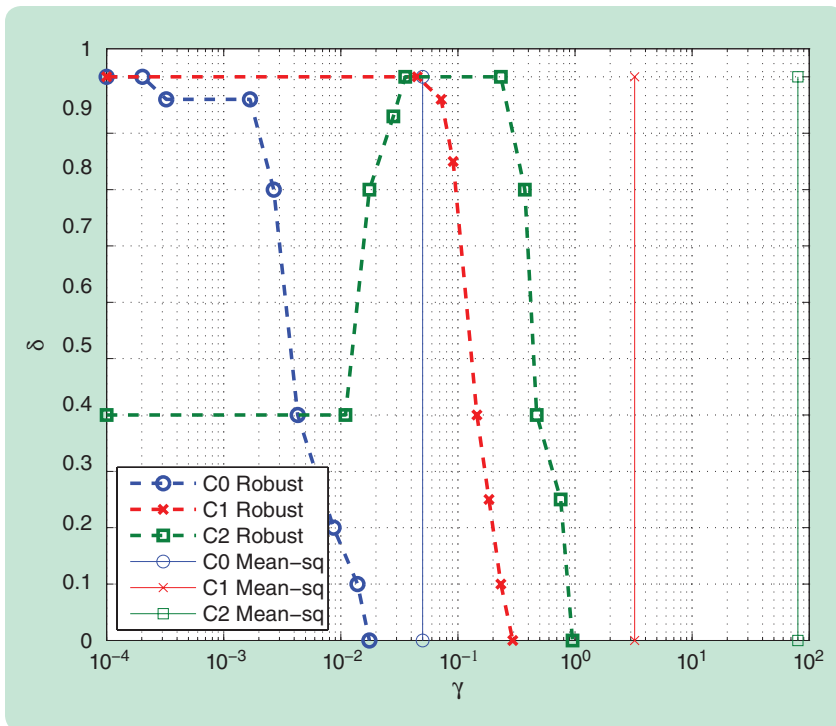


**FIGURE S5** A graphical interface used to select a networked control system analysis technique. The top section lists the different overapproximation techniques implemented for the discrete-time framework, and the bottom section lists the implemented technique for the hybrid system framework.

can be tuned for robustness in the wireless setting. For this analysis, both the linear matrix inequality (LMI) conditions [4] for the stochastic stability analysis and the robust analysis [21] (which are both implemented in the toolbox) are used to



**FIGURE 12** Polygons  $\Omega_\delta$  in the  $(h, \tau)$ -space corresponding to different values of  $\delta$ . Each polygon represents a bound on the allowable combinations of  $(h_k, \tau_k)$  and identifies an uncertainty region to consider for the robust stability analysis.



**FIGURE 13** The tradeoff between the degree of controller aggressiveness  $\gamma$  and the degree of a networked control system robustness  $\delta$  for three different controllers using the robust stability analysis (dashed lines) on the polygons  $\Omega_\delta$  and the stochastic stability analysis on the histogram in Figure 10.

verify when stochastic (mean square) and robust stability, respectively, can be guaranteed.

First, the robust stability analysis technique [21] is applied, where the investigation is on the “probability distribution free” case when the time-varying and uncertain transmission intervals and transmission delays are taken from a bounded set, without presuming any knowledge on the particular probability distribution. More information regarding these probability distribution free techniques can be found in “Robust Stability Analysis.” The objective is to quantify how much robustness can be guaranteed for different values of  $\gamma$  [in (12)]. To do so, the information in Figure 10 is used so that regions in the  $(h, \tau)$ -space can be identified for analysis purposes. The parameter  $\delta \in [0, 1]$  indicates a region  $\Omega_\delta$  that contains a subset of measurement points in Figure 10. This set  $\Omega_\delta$  contains  $\delta \times 100\%$  of points that are most likely to occur. These polygons  $\Omega_\delta$ ,  $\delta \in [0, 1]$ , are shown in Figure 12. Hence, a large value of  $\delta \in [0, 1]$  indicates that the controller is robustly stable for varying  $(h_k, \tau_k)$ -values,  $k \in N$ , in a larger set  $\Omega_\delta$  with  $\delta = 1$  corresponding to an overall closed-loop model that is robust for the complete polytope containing all measured  $(h, \tau)$ -pairs. The key aspects that are crucial for a fair comparison between the three controller structures are that 1) each value of  $\delta$  corresponds to only one polygon  $\Omega_\delta$ , and 2)  $\Omega_{\delta_1} \subseteq \Omega_{\delta_2}$  for all  $0 \leq \delta_1 \leq \delta_2 \leq 1$ . The second property ensures that, if robust stability is guaranteed for  $\delta_2$ , then robust stability is guaranteed for all  $\delta_1$  where  $0 \leq \delta_1 \leq \delta_2$ .

With such  $\delta$ -parameterization of the uncertainty region in  $(h, \tau)$ , the sufficient robust stability test in the NCS toolbox can be ran iteratively to determine the maximum  $\delta$  for which robust stability can be guaranteed for different values of  $\gamma$ , that is, different performance characteristics. The robust stability tradeoff plots were generated using the command-line functionality provided with the prototype toolbox.

Using the sufficient test for robust stability [21], the stability regions in terms of the maximum  $\delta$  for different values of  $\gamma$  are indicated by dashed lines in Figure 13 for the controllers C0, C1, and C2. This plot indicates that to be fully robustly stable, that is,  $\delta = 1$ , the controller C0 must use a very low aggressiveness  $\gamma = 0.0001$  whereas the C1 is able to be much more aggressive, as  $\gamma = 0.041$  still results in a robustly stable NCS. Increasing the aggressiveness,  $\gamma$ , above these values reduces the region,  $\Omega_\delta$ , for which robust stability of the



**The software tools support a multidisciplinary design approach  
for WCSs in which both network and control specifications are jointly  
optimized in an iterative fashion.**

NCS can be guaranteed. Interestingly, the robustness plot associated with the switched observer-based controller C2 significantly differs from those of C0 and C1. In this case, the  $\delta$  plot is not monotonically decreasing as a function of  $\gamma$ . This plot is expected to be different from C0 and C1 since C2 corrects its state estimate in a fundamentally different manner than C0 and C1. However, the particular shape of the robustness-plot of C2 is not intuitive as a consequence of the highly complex nature of the underlying NCS dynamics. Consequently, a robust stability analysis such as performed to obtain Figure 13 is necessary to gain these insights, and to properly tune the controller C2.

Next, the stochastic stability analysis technique [4] is applied for the case where time-varying and uncertain transmission intervals and transmission delays are taken from a finite set of network measurements points. The probability distribution of the set is obtained by normalizing the network measurement histogram shown in Figure 10. This analysis allows the determination of the largest value of  $\gamma$ , above which mean-square stability can no longer be guaranteed.

Using the necessary and sufficient test for mean-square stability [4], the maximum  $\gamma$  that still provides mean-square stability using the histogram in Figure 10 is indicated by solid lines in Figure 13. It is apparent that, for all three controllers, the mean-square stability results indicate larger values of  $\gamma$  can be used reliably compared to the results given by the robust stability technique. This result is expected since the robust stability analysis does not assume a probability distribution on the polygon that bounds the transmission intervals and delays. In fact, robust stability analysis allows any variation of the values for  $(h_k, \tau_k)$  in the corresponding region. Robust stability analysis techniques, such as the one used in this article, are suitable in situations when the probability distribution is hard to accurately describe (for example, due to the distribution being time-varying) or when nonstochastic phenomena determine the actual delays and transmission intervals that occur. The stochastic stability analysis technique, on the other hand, explicitly takes into account a particular distribution such as shown in Figure 10. Stochastic stability analysis techniques, such as the one used in this article, are suitable in situations when the probability distribution can be accurately described and does not change over time. Finally, it is important to keep in mind that stochastic stability considers convergence in a mean-squared sense, which, by itself, reflects a more lenient type of stability than robust stability.

## EXPERIMENTAL EVALUATION

In this section, the controllers C0, C1, and C2 are implemented on the experimental setup, and the theoretical results leading to the results in Figure 13 are validated. First a certain value of  $\gamma$  is chosen based on the robust and stochastic stability analysis results in Figure 13. The controllers corresponding to this value of  $\gamma$  are compared in terms of performance. Then the “true” stability boundary is experimentally determined and compared to the robust and stochastic stability analysis results in Figure 13.

### Controller Performance Comparison

This section experimentally evaluates the controllers C0, C1, and C2. The controllers using the largest value of  $\gamma$  corresponding to  $\delta = 0.8$  indicated in Figure 13 are compared. Specifically, the gains for C0 (using  $\gamma = 0.003$ ) are

$$K = [4.42 \quad -38.75 \quad 10.42 \quad -7.26 \quad 0.00],$$

$$L = \begin{bmatrix} 0.37 & 0.42 & 2.26 & 4.65 \\ 0.42 & 1.20 & 4.47 & 11.93 \end{bmatrix}^T,$$

the gains for C1 (using  $\gamma = 0.10$ ) are

$$K = [22.87 \quad -79.90 \quad 24.26 \quad -15.47 \quad -0.50],$$

$$L = \begin{bmatrix} 0.35 & 0.39 & 2.04 & 4.13 \\ 0.39 & 1.12 & 3.98 & 10.77 \end{bmatrix}^T,$$

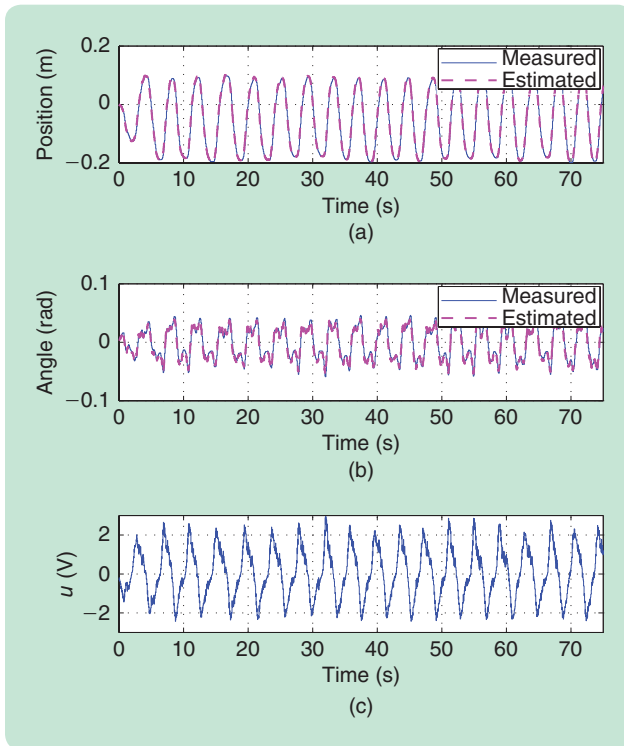
and gains for C2 (using  $\gamma = 0.37$ ) are

$$K = [39.94 \quad -114.75 \quad 35.65 \quad -21.87 \quad -0.63],$$

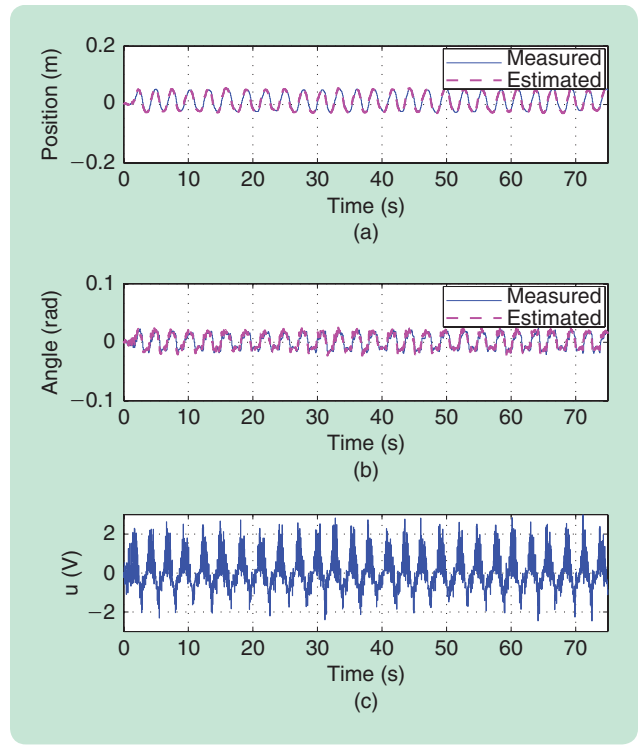
$$L_1 = [0.72 \quad 1.07 \quad 4.52 \quad 10.23]^T,$$

$$L_2 = [0.45 \quad 1.30 \quad 3.61 \quad 10.38]^T.$$

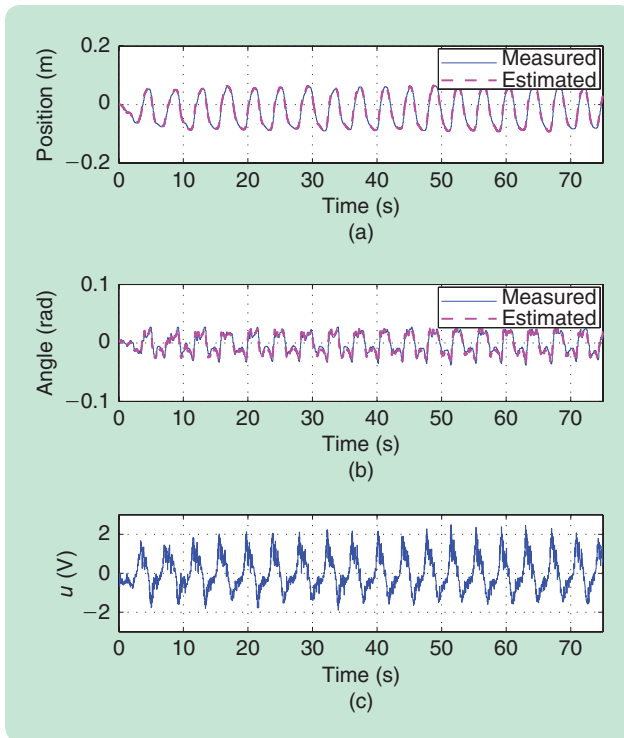
Comparing Figures 14 and 15, it can be seen that the C0 controller results in worse performance than the C1 controller because C1 is able to keep both the angle of the pendulum and cart position within a narrower set of margins around the origin. This difference in performance is attributed to the fact that, for the same degree of robustness ( $\delta = 0.8$ ), C1 allows for a higher value of  $\gamma$  than C0, which agrees with the intuition that including nominal delay information (as done for C1) improves the robustness and performance. Next, looking at Figure 16, it can be seen that, by additionally including knowledge of the sensor switching into the design, the wireless response closely resembles that of the baseline reference response (shown in Figure 11).



**FIGURE 14** Experimental measurements of controller performance using C0 with  $\gamma = 0.003$  and wireless sensor communication.



**FIGURE 16** Experimental measurements of controller performance using C2 with  $\gamma = 0.37$  and wireless sensor communication.



**FIGURE 15** Experimental measurements of controller performance using C1 with  $\gamma = 0.10$  and wireless sensor communication.

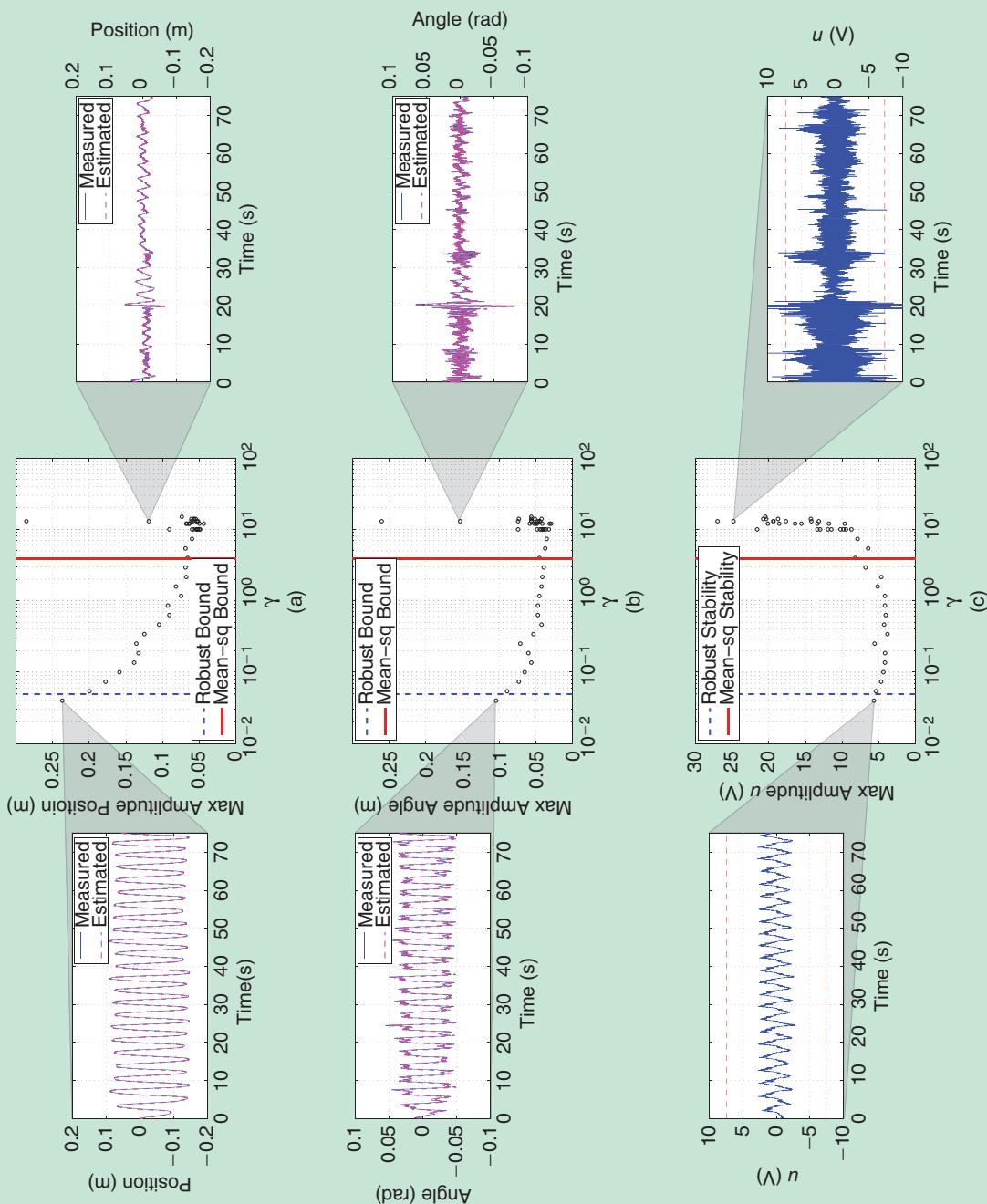
However, one noticeable difference between Figures 11 and 16 is the amount of additional high-frequency content on the input signal,  $u$ ; see Figure 16. This unwanted

high-frequency content present on the input signal for Figure 16 is actually occurring at the frequency of the controller packet reception and can be attributed to the switching of the observer. In fact, this “chattering” oscillation amplifies when the cart is disturbed away from the origin, causing the cart to shake rather violently. Therefore, it is important to realize that, although the robust stability analysis used here is very useful since certain robust stability guarantees can be provided, additional care must be taken with respect to performance properties of that solution.

### Validating the Robust Stability Boundary

In this section, several experiments are conducted to experimentally determine the “true” robust stability boundary of the wireless control setup when using controller C1. In this manner, the theoretical results obtained using the NCS toolbox, as displayed in Figure 13, can be validated. To do so, the performance of C1 at several values of  $\gamma$  is evaluated by plotting the maximum amplitude of the cart position, angle, and controller output.

The results of the experiments *that did not exhibit instability*, along with the robust stability bound for  $\Omega_1$  and the mean-square stability bound shown in Figure 13, are shown in Figure 17. Figure 17(a) and (b) shows that increasing the controller aggressiveness  $\gamma$  causes both the maximum amplitude of the cart position and the maximum amplitude of the angle to decrease until  $\gamma = 10$ . This behavior is expected since increasing the controller aggressiveness causes the controller to more effectively suppress the



**FIGURE 17** The results of conducting a series of wireless control experiments. The maximum amplitude of the (a) measured position, (b) measured pendulum angle, and (c) controller output are plotted for C1 evaluated at different values of  $\gamma$ . Time-domain signals for an experiment with  $\gamma = 0.04$  and an experiment with  $\gamma = 13$  are shown on the left and right, respectively. The robust stability bound for  $\Omega_1$  and the mean-square stability bound are also shown by the vertical lines. Only experiments that did not result in instability are shown.

steady-state oscillation caused by the nonlinear friction forces discussed in the section “Reference Behavior Under Ideal Communication.” For  $\gamma > 10$ , however, the cart itself exhibits strong vibrations, and the maximum amplitude of both the cart position and angle increase. To understand why the amplitude of both the cart position and angle increase, the control input signal, shown in Figure 17(c), needs to be investigated. By studying Figure 17(c), it can be seen that, for  $\gamma \approx 10$ , the input signal has a significant amount of noise as well as a few sequences of high ampli-

tude spikes. Due to their spontaneous occurrence, such sequences of high amplitude spikes are likely to be caused by short destabilizing sequences of transmission intervals and delays. Hence, signal quality, like that shown to the right of Figure 17(c), is an indication that the controller is too aggressive and, as such, is vulnerable to instability caused by disturbances (such as data-rate variations).

In fact, for  $\gamma > 12$ , some experiments (which cannot be plotted in Figure 17) did result in instability of the wireless control setup. In these experiments, the cart eventually did

**The NCS community has provided a multitude of different valuable theoretical techniques to determine, to different extents, which properties a particular NCS configuration possesses.**

not respect the limits of the track after initially exhibiting stable behavior (for more than 10 s). The fact that the system initially exhibited stable behavior indicates that the instability was caused by a destabilizing sequence of transmission intervals and delays. These unstable experiments, along with the data shown in Figure 17, indicates that instability occurs for  $\gamma > 12$  and the cause of instability was the network-induced effects that are present when using wireless communication for feedback control.

Several conclusions can be drawn that link the theory used for analysis to these experimental results. First, it can be seen from Figure 17 that the mean-square stability analysis bound, with its ability to incorporate a probability distribution of the transmission intervals and delays, has provided a reasonably close estimation of the true stability boundary. The NCS models in the toolbox and the probability distribution in Figure 10 form a sufficiently accurate representation of the practical NCS setup. Second, it is clear that the robust stability analysis, which does not assume any probability distribution of the transmission intervals and delays, has provided a more conservative bound than the true stability boundary. This bound does, however, provide the added assurance that stability is still guaranteed if the probability distribution (within  $\Omega_1$ ) changes over time. Third, using these techniques in combination has indicated a range of controller aggressiveness values, between which different degrees of robustness are guaranteed. Both bounds are useful because, in general, the particular degree of required robustness depends on the application. Hence, being able to conveniently apply both of these techniques in practice can provide insights into controller design as well as demonstrates the benefits of having multiple NCS analysis and synthesis techniques available in the form of software tools, which can be used in an easy manner.

## CONCLUSION

The focus of this article was on the analysis, design, and experimental validation of WCSs, which form a class of CPSs that exhibit physical, control, computation, and communication components. The effectiveness of theoretical NCS results implemented in an NCS toolbox was verified on a wireless control case study for an inverted plant/cart system. The focus of the toolbox was on robust and stochastic stability analysis, which is a well-developed topic in the NCS literature. Performance analysis and controller design tools are far less mature and more work in this area is needed, before implementation in the toolbox is pursued.

The design of robustly stabilizing observer-based controllers was based on appropriately choosing weights in LQR/LQE cost functions in a nominal setting and subsequently using the prototype NCS toolbox to aid in tuning the controller to achieve robustness and performance in the wireless setting. The NCS toolbox was used to generate both robust and mean-square stability tradeoff-versus-performance plots (for two different controller structures) that provide the theoretical guarantees needed to aid in tuning such controllers. Applying both the robust and mean-square stability analysis techniques provided quantitative design guidelines for guaranteeing different degrees of robustness. These theoretical bounds were compared with the true stability bounds that were determined by conducting a series of wireless control experiments. The theoretical mean-square stability boundaries were reasonably close to the experimentally determined bounds, which showed the accuracy and relevance of the NCS models in the toolbox.

Based on the conducted wireless control experiments comparing two different controller structures, another interesting observation was made. The optimal controller structure that seems theoretically most suited (and performed the best) exhibited an undesirable chattering oscillation at the transmission frequency due to switching observer gains, which rendered the controller less suitable in practice. This phenomenon, along with several other observations made in this article, indicate future challenges that need to be addressed. The hope is that these findings encourage others to make their theoretical techniques available through software tools and conduct similar wireless control investigations to help pave an experimentally driven path to identify the most promising research directions in the NCS field so that theoretical advances can be geared accordingly.

## ACKNOWLEDGMENTS

This work is supported by the Innovational Research Incentives Scheme under the VICI grant “Wireless control systems: A new frontier in automation” (no. 11382) awarded by The Netherlands Organization for Scientific Research, and the Dutch Science Foundation, by the European 7th Framework Network of Excellence “Highly complex and networked control systems (HYCON2),” under program 257462.

## AUTHOR INFORMATION

*Nicolas W. Bauer* (nicolaswilliam.bauer@gmail.com) received the B.Sc. with honors and the M.Sc. from the



Department of Electrical and Computer Engineering, University of California, Santa Barbara, in 2007 and 2008, respectively. He received his Ph.D. from the Department of Mechanical Engineering at Eindhoven University of Technology, The Netherlands, in February 2013. His current research interests include networked control systems, distributed control, and switched systems. He can be contacted at Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

**S.J.L.M. (Bas) van Loon** received the M.Sc. with great appreciation in mechanical engineering in 2011 from Eindhoven University of Technology, The Netherlands, where he is currently pursuing a Ph.D. degree. His current research interests include networked control systems, reset control and modeling, and control of hybrid and switched systems.

**Nathan van de Wouw** obtained the M.Sc. with honors and the Ph.D. in mechanical engineering from Eindhoven University of Technology, The Netherlands, in 1994 and 1999, respectively. From 1999 until 2014, he has been affiliated with the same department where he is currently an associate professor. He also holds an adjunct professor position at the University of Minnesota. He worked at Philips Applied Technologies, Eindhoven, The Netherlands, in 2000 and at the Netherlands Organisation for Applied Scientific Research, Delft, in 2001. He was a visiting professor at the University of California, Santa Barbara, in 2006–2007, at the University of Melbourne, Australia, in 2009–2010, and at the University of Minnesota, in 2012 and 2013. He has published many journal and conference papers and the books *Uniform Output Regulation of Nonlinear Systems: A Convergent Dynamics Approach* with A.V. Pavlov and H. Nijmeijer (Birkhäuser, 2005) and *Stability and Convergence of Mechanical Systems with Unilateral Constraints* with R.I. Leine (Springer-Verlag, 2008). He is an associate editor for *Automatica* and *IEEE Transactions on Control Systems Technology*. His research interests are the analysis and control of nonlinear/nonsmooth systems and networked control systems.

**W.P.M.H. (Maurice) Heemels** received the M.Sc. in mathematics and the Ph.D. in control theory (both *summa cum laude*) from Eindhoven University of Technology (TU/e), The Netherlands, in 1995 and 1999, respectively. After being an assistant professor in the Department of Electrical Engineering at TU/e and a research fellow at the Embedded Systems Institute, he is currently a full professor in the Control Systems Technology Group in the Department of Mechanical Engineering at TU/e. He held visiting research positions at the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2001, at Océ, Venlo, The Netherlands, in 2004, and at the University of California, Santa Barbara, in 2008. He is an associate editor for *Nonlinear Analysis: Hybrid Systems* and *Automatica*. He was the recipient of a VICI grant of the Dutch Technology Foundation and The Netherlands Organisation for Scientific Research.

His current research interests include general systems and control theory, hybrid and cyberphysical systems, networked and event-triggered control, and constrained systems including model predictive control.

## REFERENCES

- [1] A. A. Alam, A. Gattami, and K. H. Johansson, "Suboptimal decentralized controller design for chain structures: Applications to vehicle formations," in *Proc. IEEE Conf. Decision Control European Control Conf.*, 2011, pp. 6894–6900.
- [2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. New York: Dover, 2005.
- [3] P. J. Antsaklis, M. J. McCourt, H. Yu, P. Wu, and F. Zhu, "Cyber-physical systems design using dissipativity," in *Proc. 31st Chinese Control Conf.*, 2012, pp. 1–5.
- [4] D. Antunes, J. P. Hespanha, and C. Silvestre, "Stochastic networked control systems with dynamic protocols," *Asian J. Control*, Feb. 2014. to be published.
- [5] M. Athans, "The role and use of the stochastic linear-quadratic-Gaussian problem in control system design," *IEEE Trans. Automatic Contr.*, vol. 16, no. 6, pp. 529–552, Dec. 1971.
- [6] N. W. Bauer, P. J. H. Maas, and W. P. M. H. Heemels, "Stability analysis of networked control systems: A sum of squares approach," *Automatica*, vol. 48, no. 8, pp. 1514–1524, 2012.
- [7] N. W. Bauer, S. J. L. M. van Loon, M. C. F. Donkers, N. van de Wouw, and W. P. M. H. Heemels, "Networked control systems toolbox: Robust stability analysis made easy," in *Proc. 3rd IFAC Workshop Distributed Estimation Control Networked Systems*, June 2012, pp. 55–60.
- [8] A. Bemporad, W. P. M. H. Heemels, and M. Johansson, Eds., *Networked Control Systems*, vol. 406. London: Springer-Verlag, 2010.
- [9] S. Bittanti and P. Colaneri, *Periodic Systems: Filtering and Control*. London: Springer-Verlag, 2009.
- [10] F. Blaabjerg, R. Teodorescu, M. Liserre, and A. V. Timbus, "Overview of control and grid synchronization for distributed power generation systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1398–1409, Oct. 2006.
- [11] D. Carnevale, A. R. Teel, and D. Nešić, "A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems," *IEEE Trans. Autom. Contr.*, vol. 52, no. 5, pp. 892–897, May 2007.
- [12] G. Cembrano, G. Wells, J. Quevedo, R. Perez, and R. Argelaguet, "Optimal control of a water distribution network in a supervisory control system," *Control Eng. Pract.*, vol. 8, no. 10, pp. 1177–1188, 2000.
- [13] A. Cervin, D. Henriksson, B. Lincoln, K. E. J. Eker, and Årzen, "How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime," *IEEE Control Syst. Mag.*, vol. 23, no. 3, pp. 16–30, June 2003.
- [14] O. Chipara, C. Lu, T. C. Bailey, and G. Roman, "Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit," in *Proc. ACM Conf. Embedded Networked Sensor Systems*, New York, 2010, pp. 155–168.
- [15] M. B. G. Cloosterman, L. Hetel, N. van de Wouw, W. P. M. H. Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked control systems," *Automatica*, vol. 46, no. 10, pp. 1584–1594, 2010.
- [16] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Trans. Autom. Contr.*, vol. 54, no. 7, pp. 1575–1580, July 2009.
- [17] R. Cunha, A. Severino, N. Pereira, A. Koubaa, and M. Alves, "Zigbee over TinyOS: Implementation and experimental challenges," in *Proc. 8th Portuguese Conf. Automatic Control*, Apr. 2008, pp. HURRAY-TR-090202.
- [18] D. F. Delchamps, "Stabilizing a linear system with quantized state feedback," *IEEE Trans. Autom. Contr.*, vol. 35, no. 8, pp. 916–924, Aug. 1990.
- [19] X. C. Ding, M. Powers, M. Egerstedt, S.-Y. Young, and T. Balch, "Executive decision support," *IEEE Robot. Automat. Mag.*, vol. 16, no. 2, pp. 73–81, 2009.
- [20] M. C. F. Donkers, W. P. M. H. Heemels, D. Bernardini, A. Bemporad, and V. Shneer, "Stability analysis of stochastic networked control systems," *Automatica*, vol. 48, no. 5, pp. 917–925, 2012.
- [21] M. C. F. Donkers, W. P. M. H. Heemels, N. van de Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Trans. Autom. Contr.*, vol. 56, no. 9, pp. 2101–2115, Sept. 2011.

- [22] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," in *Proc. SPIE*, 2005, vol. 5765, pp. 477–484.
- [23] J. Eker, A. Cervin, and A. Hörjel, "Distributed wireless control using Bluetooth," in *Proc. IFAC Conf. New Technologies Computer Control*, Nov. 2001, pp. 347–352.
- [24] N. Elia and S. K. Mitter, "Stabilization of linear systems with limited information," *IEEE Trans. Autom. Contr.*, vol. 46, no. 9, pp. 1384–1400, 2001.
- [25] U. Fechner and R. Schmehl, "Design of a distributed kite power control system," in *Proc. IEEE Int. Conf. Control Applications*, 2012, pp. 800–805.
- [26] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2006.
- [27] H. Fujioka, "Stability analysis for a class of networked/embedded control systems: A discrete-time approach," in *Proc. American Control Conf.*, June 2008, pp. 4997–5002.
- [28] R. H. Gielen, S. Oлару, M. Lazar, W. P. M. H. Heemels, N. van de Wouw, and S.-I. Niculescu, "On polytopic inclusions as a modeling framework for systems with time-varying delays," *Automatica*, vol. 46, no. 3, pp. 615–619, 2010.
- [29] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, no. 2, pp. 28–93, Apr. 2009.
- [30] E. Guizzo, "Three engineers, hundreds of robots, one warehouse," *IEEE Spectr.*, vol. 45, no. 7, pp. 26–34, 2008.
- [31] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nešić, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance," *IEEE Trans. Autom. Contr.*, vol. 55, no. 8, pp. 1781–1796, Aug. 2010.
- [32] W. P. M. H. Heemels, N. van de Wouw, R. H. Gielen, M. C. F. Donkers, L. Hetel, S. Oлару, M. Lazar, J. Daafouz, and S. Niculescu, "Comparison of over-approximation methods for stability analysis of networked control systems," in *Proc. Int. Conf. Hybrid Systems: Computation Control*, 2010, pp. 181–190.
- [33] J. J. Hench and A. J. Laub, "Numerical solution of the discrete-time periodic Riccati equation," *IEEE Trans. Autom. Contr.*, vol. 39, no. 6, pp. 1197–1210, 1994.
- [34] A. Hernández, "Wireless inverted pendulum using IEEE 802.15.4 protocol," M.S. thesis, KTH Royal Inst. Technol., Stockholm, Sweden, Apr. 2011.
- [35] J. P. Hespanha, *Linear System Theory*. Princeton, NJ: Princeton Univ. Press, 2009.
- [36] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
- [37] A. Ilzhöfer, B. Houska, and M. Diehl, "Nonlinear MPC of kites under varying wind conditions for a new class of large-scale wind power generators," *Int. J. Robust Nonlinear Control*, vol. 17, no. 17, pp. 1590–1599, 2007.
- [38] K. E. Johnson and N. Thomas, "Wind farm control: Addressing the aerodynamic interaction among wind turbines," in *Proc. American Control Conf.*, June 2009, pp. 2104–2109.
- [39] B. Kosucu, K. Irgan, G. Kucuk, and S. Baydere, "FireSenseTB: A wireless sensor networks testbed for forest fire detection," in *Proc. Int. Conf. Wireless Communications Mobile Computing: Connecting World Wirelessly*, New York, 2009, pp. 1173–1177.
- [40] D. S. Laila, D. Nešić, and A. R. Teel, "Open and closed-loop dissipation inequalities under sampling and controller emulation," *Eur. J. Control*, vol. 8, no. 2, pp. 109–125, 2002.
- [41] E. A. Lee, "CPS foundations," in *Proc. ACM/IEEE Design Automation Conf.*, 2010, pp. 737–742.
- [42] E. A. Lee and S. A. Seshia. (2011). Introduction to embedded systems, A cyber-physical systems approach. [Online]. Available: <http://LeeSeshia.org>
- [43] C.-H. Loh, J. P. Lynch, K.-C. Lu, Y. Wang, C.-M. Chang, P.-Y. Lin, and T.-H. Yeh, "Experimental verification of a wireless sensing and control system for structural control using MR dampers," *Earthquake Eng. Struct. Dyn.*, vol. 36, no. 10, pp. 1303–1328, 2007.
- [44] S. J. L. M. van Loon, M. C. F. Donkers, N. van de Wouw, and W. P. M. H. Heemels, "Stability analysis of networked and quantized linear control systems," *Nonlinear Anal.: Hybrid Syst.*, vol. 10, no. C, pp. 111–125, 2013.
- [45] Jennic, "Application note: Coexistence of IEEE 802.15.4 at 2.4 GHz," Jennic Ltd, Sheffield, South Yorkshire, Tech. Rep. JN-AN-1079, 2008.
- [46] Y. Ma, M. Richards, M. Ghanem, Y. Guo, and J. Hassard, "Air pollution monitoring and mining based on sensor grid in London," *Sensors*, vol. 8, no. 6, pp. 3601–3623, 2008.
- [47] L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Trans. Autom. Contr.*, vol. 49, no. 9, pp. 1562–1572, Sept. 2004.
- [48] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proc. IEEE*, vol. 95, no. 1, pp. 29–47, 2007.
- [49] P. Naghshtabrizi and J. P. Hespanha, "Stability of network control systems with variable sampling and delays," in *Proc. Annu. Allerton Conf. Communication, Control, Computing*, 2006, pp. 1328–1335.
- [50] D. Nešić and A. R. Teel, "Input-output stability properties of networked control systems," *IEEE Trans. Autom. Contr.*, vol. 49, no. 10, pp. 1650–1667, 2004.
- [51] D. Nešić and A. R. Teel, "Input-to-state stability of networked control systems," *Automatica*, vol. 40, no. 12, pp. 2121–2128, 2004.
- [52] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Intell. Transp. Syst. Trans. Mag.*, to be published.
- [53] S. Öncü, N. van de Wouw, W. M. P. H. Heemels, and H. Nijmeijer, "String stability of interconnected vehicles under communication constraints," in *Proc. IEEE Conf. Decision Control*, Dec. 2012, pp. 2459–2464.
- [54] M. Petrova, J. Riihijärvi, P. Mähönen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proc. IEEE Wireless Communications Networking Conf.*, Apr. 2006, vol. 1, pp. 487–492.
- [55] J. Ploeg, A. F. A. Serrarens, and G. J. Heijenk, "Connect and drive: Design and evaluation of cooperative adaptive cruise control for congestion reduction," *J. Mod. Transp.*, vol. 19, no. 3, pp. 207–213, 2011.
- [56] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 58–71, June 2004.
- [57] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proc. Int. Symp. Information Processing Sensor Networks*, Apr. 2005, pp. 364–369.
- [58] Quanser, "Single inverted pendulum instructor manual," Quanser Inc., Markham, ON, Tech. Rep. 513 r4.1.
- [59] J. R.-d. Solar, E. Chown, and P. G. Ploeger, Eds., *RoboCup 2010: Robot Soccer World Cup XIV* (Lecture Notes in Computer Science), vol. 6556. Berlin Heidelberg, Germany: Springer-Verlag, 2011.
- [60] L. Schenato, "To zero or to hold control inputs with lossy links?" *IEEE Trans. Autom. Contr.*, vol. 54, no. 5, pp. 1093–1099, 2009.
- [61] J. J. C. van Schendel, M. C. F. Donkers, W. P. M. H. Heemels, and N. van de Wouw, "On dropout modelling for stability analysis of networked control systems," in *Proc. American Control Conf.*, July 2010, pp. 555–561.
- [62] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proc. American Control Conf.*, 2001, vol. 2, pp. 1491–1496.
- [63] P. Seiler and R. Sengupta, "An  $H_\infty$  approach to networked control," *IEEE Trans. Autom. Contr.*, vol. 50, no. 3, pp. 356–364, Mar. 2005.
- [64] Y. Shi and B. Yu, "Output feedback stabilization of networked control systems with random delays modeled by Markov chains," *IEEE Trans. Autom. Contr.*, vol. 54, no. 7, pp. 1668–1674, July 2009.
- [65] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Contr.*, vol. 49, no. 9, pp. 1453–1464, Sept. 2004.
- [66] J. Stipanovits, X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, and S. Wang, "Toward a science of cyber-physical system integration," *Proc. IEEE*, vol. 100, no. 1, pp. 29–44, 2012.
- [67] N. van de Wouw, P. Naghshtabrizi, M. B. G. Cloosterman, and J. P. Hespanha, "Tracking control for sampled-data systems with uncertain time-varying sampling intervals and delays," *Int. J. Robust Nonlinear Contr.*, vol. 20, no. 4, pp. 387–411, 2010.
- [68] N. van de Wouw, D. Nešić, and W. P. M. H. Heemels, "A discrete-time framework for stability analysis of nonlinear networked control systems," *Automatica*, vol. 48, no. 6, pp. 1144–1153, 2012.
- [69] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems*, New York, 2004, pp. 13–24.
- [70] L. Yuxin, W. Jianqiang, M. Tao, and S. Gengsu, "Study on experimental table wireless control system of open laboratory based on Zig-Bee technology," in *Proc. Int. Conf. Computer Science Education*, 2009, pp. 1396–1399.
- [71] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953–958, 2006.
- [72] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst.*, vol. 21, no. 1, pp. 84–99, Feb. 2001.

