Effects of Saturation for High-Throughput Satellite Buses

Carvajal Godínez, Johan; Guo, Jian; Gill, Eberhard

DOI
10.1109/TAES.2019.2940341

Publication date
2019

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Aerospace and Electronic Systems

Effects of Saturation for High-Throughput Satellite Buses

Johan Carvajal-Godinez, Jian Guo and Eberhard Gill

***Abstract*—Bus saturation is deemed as one of the primary causes of delays in the data propagation between spacecraft components. However, the conditions under which bus congestion can affect measurements variability were not well characterized before. This work presents a bus saturation model and a set of experiments to characterize the bus performance of satellite missions for different traffic load, data rate, and synchronization periods. The results showed an increase of measurements variance of up to 18% caused by bus saturation. Additionally, an algorithm was proposed to reduce data delay by controlling saturation on the communication channel at operational level.**

***Keywords*—Bus Saturation, Measurement Delay, Measurement Variance, CAN protocol, AOCS, FDIR**

I. Introduction

The complexity of spacecraft buses is increasing as a result of higher processing demands on space missions requirements. Take, for instance, the attitude and orbit control subsystem (AOCS) of a satellite for high-speed optical communication. The required pointing accuracy for demonstrating multiple high-speed laser communication links is in the range of 1 μrad to 1 mrad [1], compared to the 10 to 100 mrad required for satellites using radio-frequency links [2].

The most effective mechanism to increase processing performance on satellites is distributing functions at the components level. The distribution of tasks generates inter-operational networks with higher processing performance, but it comes at the cost of increasing the risk of data delay.

Communication in distributed sensor networks has been widely researched for establishing strategies to balance data load and processing power [3]-[4]. In [5], characteristics of sensor and actuator networks on satellites are discussed along with strategies to deal with the latency of data propagation. It is clear that decentralized control achieves more benefits on reliability and performance when comparing it with a centralized processing architecture [6].

Within the spacecraft, the AOCS can be described as a networked command and control system [7]. That implies the use of distributed communication links to interconnect sensors, actuators and the AOCS computer with the rest of the spacecraft to perform periodic tasks. One critical aspect of command and control systems is their real-time nature [8]. That makes AOCS very susceptible to delays in data propagation on its internal communication bus.

Johan Carvajal-Godinez, Jian Guo and Eberhard Gill are with the Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands. E-mail: j.guo@tudelft.nl

Manuscript received TBD; revised TBD.

Delay in data propagation leads to errors in satellites operation, for instance, it can produce under-or-over actuation during spacecraft's maneuvers. It can also cause a reduction in the stability margins of controllers, leading to undesirable operation conditions [9]-[10]. Faulty communication channels can also produce data loss during spacecraft state estimation, which might be reflected on data fusion issues [11] or higher latency for parameters update in the AOCS computer.

The cause of delay in data propagation can be attributed either to physical phenomena like electromagnetic interference in the channel, or to human-related issues in the bus design and implementation [12].

There are several techniques to address fault detection and recovery for the implementation of distributed communication networks. In general, four kinds of faults can be identified in distributed communication systems [13]. These are (1) data corruption, (2) channels disconnection, (3) loss of synchronization and (4) channels saturation.

Usually, three implementation parameters can be controlled to mitigate channel saturation [14]. These are data rate, channel synchronization period and channel's buffer size. For implementation purposes, it is also necessary to select a communication protocol that has proven high reliability in harsh environments.

The Controller Area Network (CAN) protocol has been adopted in automotive and industrial applications for more than 30 years due to its environment resilience characteristics [15]-[16]. Currently, there are development efforts to get hardware, firmware, and software in place to enable CAN for spacecraft onboard communications and control systems, especially allowing integrated fault detection and correction capabilities, as well as time-triggered mechanisms for scheduling message transmission over a communication cycle [17].

On the algorithm side, most of the estimation techniques for AOCS are based on Kalman filters. The filter's measurement model usually neglects the effects of delayed measurements, or it tries to accommodate them on its implementation rather than in its design. In [18] a distributed estimation algorithm is proposed to deal with delays in networked sensors systems. The main disadvantage of this approach is its susceptibility to communication reliability and matrices dimensions, that can generate adverse conditions on the navigation computer.

Bar-Shalom [19] proposes and compares three algorithms for giving an exact solution to the out-of-order measurement sequences created by delays in the communication channel. This approach requires storing of the latest state for using it in case of a missing measurement. The main drawback of this method is the processing overhead in the estimation computer, which may pose a risk for performance.

This paper studies implementation aspects of a distributed communication architecture for AOCS in small satellites that have not been explored before, to find a link between channel saturation and pointing error performance. The focus is on characterizing the effects of saturation due to high-throughput operation scenarios. These results are used later to understand the impact of measurements delay in the precision of the satellites attitude estimation.

The main contribution of this work is the relationship found between the communication channel saturation and the estimation errors due to measurements delays in distributed communications architectures. That makes this work relevant because it allows designers to perform a better trade-off process to establish mitigation plans by design. Also, the paper proposes an algorithm for saturation mitigation at an operation level. The behavior of the channel saturation is described by a bus utilization model that is introduced analytically and then verified using an AOCS case study. The approach proposed here is intended for mission-critical space systems, but it can also be applied for terrestrial applications.

The advantage of the proposed estimation approach is that instead of splitting its implementation, the measurement model is extended to account for the effects of delays as part of its measurement variability while keeping a monolithic filter architecture. That avoids adverse conditions in the estimator algorithm due to matrix size inconsistencies, and it also keeps implementation complexity fixed. At the same time, the bus utilization model proposed provides new perspectives on communication control that can be applied to improve the resilience of satellite's software.

The paper is organized as follows: Section II focuses on the problem formulation by presenting the modeling of the main components of the system. Section III describes the communication bus implementation, as well as the methodology followed to reproduce the satellite operation conditions. Section IV focuses on the experimental design and it describes case study scenarios. Results and their analysis are drawn in Section V. Finally, conclusions and further recommendations are provided in Section VI.

II. Problem Formulation

The AOCS subsystem can be described as a non-linear system in the state space as

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t)] + \mathbf{w}(t) \tag{1}$$

where, $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control vector and $\mathbf{w}(t)$ is the perturbation vector modeled as a random variable that follows a normal distribution. A set of sensors are assumed to produce a measurement vector $\mathbf{y}(t)$ at time equals t_k as a function $\mathbf{h}$ of both the state vector $\boldsymbol{x}(t)$ and the measurement errors $\mathbf{M}_E(t)$.

$$\mathbf{y}(t_k) = \mathbf{h}[\mathbf{x}(t_k)] + \mathbf{M}_E(t_k) \tag{2}$$

In (2), $\mathbf{M}_E(t_k)$ accounts for errors in the measurements, including additive Gaussian white noise.

In the system model introduced by (1), the spacecraft's dynamics are described by the change in time of angular momentum as a function of its angular velocity according to following expression

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \mathbf{T} \tag{3}$$

where, $\mathbf{J}$ is the spacecraft's moment of inertia, $\boldsymbol{\omega}$ the spacecraft's angular velocity, and $\mathbf{T}$ denotes the addition of the control and the disturbances torques affecting the spacecraft, for instance, the atmospheric drag and the magnetic field perturbations.

A. Estimation Model

The system presented in (1) needs to become linear so that an engineering model can be implemented on the AOCS onboard computer to estimate the satellite orientation. The use of an Extended Kalman Filter is proposed to predict and update the spacecraft state from a non-linear model as described in [20] and presented as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{h}[\mathbf{x}(k)] + \mathbf{v}(k) \end{aligned} \tag{4}$$

where, $\mathbf{x}(k) \in \mathbb{R}^n$ is the attitude state vector, $\mathbf{u}(k) \in \mathbb{R}^c$ is the attitude control vector, $\mathbf{y}(k) \in \mathbb{R}^L$ is the measured output from the attitude determination and control subsystem. Functions $\mathbf{f}$ and $\mathbf{h}$ are linear expressions obtained from the non-linear model using, for instance, Taylor's series expansion.

In (4), both $\mathbf{w}(k)$ and $\mathbf{v}(k) \in \mathbb{R}^n$ represent the process and the measurement noise, respectively. The engineering model assumes white noise for $\mathbf{w}(k)$ and $\mathbf{v}(k)$ described as normal distributions functions with zero mean and covariance matrices $\mathbf{Q} > 0$ and $\mathbf{R}_v > 0$, respectively. The system's model also ignores the pointing errors between the ground station and space terminal for simplifying the estimation algorithm implementation.

State estimators using distributed communication architectures are known to have fading channels that impact their performance. The measurement model is then extended to accommodate this effect as presented in [21] as follows

$$\mathbf{z}(k) = \xi_k \mathbf{y}(k) + \boldsymbol{n}_k \tag{5}$$

where, $\mathbf{n}_k \in \mathbb{R}^L$ is also additive white noise accounting for delays in the measurements, and $\xi_k \in \mathbb{R}^{LxL}$ is a diagonal matrix accounting for the fading effects on the i_{th} link of the communication bus.

This structure is introduced as

$$\begin{aligned} \xi_k &= \text{diag}\{\xi_{1,k}, \xi_{2,k}, ..., \xi_{i,k}\} \\ \xi_{i,k} &= \gamma_{i,k}\Omega_{i,k} \end{aligned} \tag{6}$$

where, $\xi_{i,k}$ represents the faults mechanisms for the communications channel. In this expression $\gamma_{i,k}$ is a Bernoulli process that models the arrival of measurements, and $\Omega_{i,k}$ describes the signal fluctuation due to channel's performance degradation as discussed in [22]. The arrival of measurements is considered as a success if the data arrives before a delay threshold, or as a failure if it exceeds the threshold or if it gets lost during its propagation as described in [23].

Using the all above's expressions, the AOCS estimation model can be re-written to include the effects of faulty communication channels as follows [24].

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] + \mathbf{w}(k) \\ \mathbf{z}(k) &= \xi_k \mathbf{h}[\mathbf{x}(k)] + \boldsymbol{n}_k + \boldsymbol{v}_k \end{aligned} \tag{7}$$

B. Channel Saturation Model

Saturation refers to the capacity of a communication channel to deal with incoming messages beyond its nominal capacity. It can be measured by monitoring the bus utilization (BU), which is directly related to the number of incoming messages and the implementation parameters of the communication channel, as well as, the network size. The bus utilization is defined as the relation between the time required to transmit a group of messages divided by the total time available in the bus to complete this task every transmission cycle. For CAN protocol an accurate value for the worst case transmission time is described in [25].

Bus utilization analysis requires accounting the total number of messages transmitted over the communication channel per synchronization event. Assuming there are N nodes sharing messages with length M_L [bits] over the communication bus, the message volume is defined by the number of messages and their length as

$$M_S = \sum_{i=1}^{N} M_{Li} \ [bits] \tag{8}$$

where, M_S is the total number of bits that are sent over the communication bus per cycle. It is assumed that the communication bus has a constant synchronization period T_{Sync} in seconds and a constant bus data rate D_R in bits per second. This assumption allows quantifying the impact of the topology configuration in the bus utilization (BU) following the following relationship:

$$BU = \frac{1}{T_{Sync} D_R} \sum_{i=1}^{N} M_{Li} + \frac{M_L}{D_R \lambda} \ [\%] \tag{9}$$

From (9), the link between bus utilization and the number of sent messages is direct. Also, note that the increase in bus utilization is dependent on the average message arrival rate λ used to describe the additional traffic injected.

It is important to keep in mind that (9) assumes the channel is operated under nominal conditions (not saturated), which means that there is enough time to transmit the total number of messages between synchronization events. That is critical to avoid losing messages with low priority.

III. Communication Bus Implementation

This section focuses on the implementation aspects of the simulation model developed to quantify the effect of propagation delays on sensor's measurements received at the AOCS computer.

A. AOCS Reference Architecture

The simulation model for the satellite intends to reproduce an AOCS reference architecture as shown in Figure 1. In this diagram, the AOCS onboard computer (OBC) is in charge of providing the processing capabilities for attitude and orbit control by means of a multi agent systems-based application running on the AOCS computer.

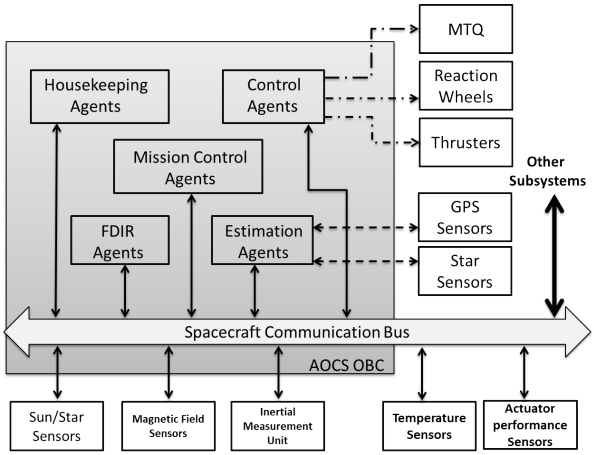

Figure 1. AOCS Reference Architecture

In Figure 1, the communication between components is divided in two categories: peer-to-peer (dotted lines) and distributed (solid lines) communication. Traditionally, peer-to-peer communication is required to interface highly complex sensors and actuators, for instance Global Positioning System (GPS) receivers and magnetic torquer (MTQ) units, whereas distributed buses are used for less complex and more abundant devices. From Figure 1, it is clear that most of the AOCS sensors are connected to the estimation agent via the spacecraft communication bus. In that figure there is a group of fault detection detection, isolation and recovery (FDIR) agents implemented at software level to improve the reliability features of the system.

Small satellites, in particular Cubesats, are more constrained in terms of volume and power. That is a motivation for implementing common interfaces for internal spacecraft communication, usually in the form of linear bus topologies. For example, Delfi-Next satellite [26], used of a distributed data bus to simplify the physical interface between multiple components and subsystems. Several other satellites take advantage of distributed communication, but this work focuses on those that have high volume and power constraints.

It was necessary to implement a simulation model using MATLAB^{TM} Simulink to properly reproduce the data delays in the spacecraft communication bus. For that purpose, the CAN protocol was selected since it is one of the most promising fault-tolerant distributed communication protocols being adopted in the micro/nano-satellite community [27]. One of the main advantages of CAN protocol is its maturity level in harsh environments, such as automotive application as discussed in [28]. There are several CAN implementations that can be

embraced for space applications. For instance, Flexible Time-Triggered (FTT) CAN combines event and time-triggered capabilities to enable flexible operation of systems [29]. The following subsections elaborate on the implementation of the simulation model for synchronous CAN protocol, as well as the implementation of the sensor model and the traffic generation model used to simulate satellite's high throughput operation scenarios.

B. Channel Implementation

The implementation of the communication channel was divided in two parts. One to describe the mechanisms to access the physical channel known as medium access control (MAC), and the second one to implement the channel controller and their interfaces with the application layer. The model was achieved by modifying a CAN model used for control of Antilock Brake System (ABS) in automobiles presented in [30]-[31].

1) Medium Access Control: Given that CAN operates using a bus topology, it is necessary to establish a method for all the nodes to access the communication medium. CAN establishes that the physical layer is specified through the standard ISO 11898-3 for datarates up to 1000 kbps. In the simulation model, the channel was implemented using a discrete time simulation approach to make it synchronous. This implementation consisted of two queues. One for receiving and ordering messages by priority using the internal message ID given by its node address, and the other to queue and broadcast the messages to the nodes connected to the bus.

In addition to the arbitration and broadcast queues used to organize messages by priority, there was a control logic block in charge of executing the medium access control algorithm to determine when a message is allowed to be processed. The process block simulates the time required by the message entity to propagate over the physical channel. This parameter is called process time delay P_T, and it is fixed as a function of the channel data rate capacity D_R, the number of bits per CAN message NB_{CF} and the channel physics (e.g., cable length) NB_P as

$$P_T = \frac{NB_{CF} + NB_P}{D_R} \ [s] \tag{10}$$

The simulation model operates using two data rates. One called baseline at the speed of 500 kbps and the high speed at 1000 kbps. The size of the CAN payload data was constant (64 bits) for a total encoding length of 113 bits per CAN message. This size included penalties for channel physics, and it excludes stuffing bits. The control logic block also included an algorithm to establish the percentage of messages lost in the channel.

2) CAN controller: On top of the physical model abstraction, a CAN controller model was synthesized for packing and unpacking data transmitted over the bus. It consisted of a transmitter and a receiver. The transmitter took the incoming data from the upper application layer and framed it into a CAN message following the standard CAN framing structure. Additionally, each message contained a time stamp that was used to calculate the delay of a message on arrival. Also, the transmitter was provided with a queue to hold messages when the communication channel was busy. The length of that queue was set to a capacity of 3 messages, to make it consistent with the buffers size of a commercial-off-the-shelf microcontroller (SMT32F405) used as reference. The transmitter also was in charge of generating the synchronization for the transmission of the messages over the bus. That was controlled during the simulation with the channel synchronization period (T_{Sync}) parameter. The scheduling of transmission and reception was assumed ideal [32].

The receiver consisted of a reception buffer with a capacity for three incoming messages from the communication channel layer. Later, a reception protocol was applied to these messages to filter them by node ID and verifying its data integrity using the CAN Cyclic Redundancy Check (CRC) specification. Then, the messages were decoded, and the payload data was retrieved and forwarded to the upper layers at the application level. Figure 2 depicts the interface of the CAN controller and the medium access control in the implementation con the communication channel.

At this point the delay Δt of an specific CAN message propagated from its source node at the sensor to the destination at the AOCS computer can be calculated as

$$\Delta t = t_S - t_{AOCS} \ [s] \tag{11}$$

where, the t_S is the time at which the data was packed into a CAN message at the source node and the t_{AOCS} is the time in which the message is unpacked at the destination node in the navigation computer. The CAN controller model required to be provided with the following parameters for its operation: CAN controller ID, payload data size, IDs of subscribed nodes and a range of valid node IDs.

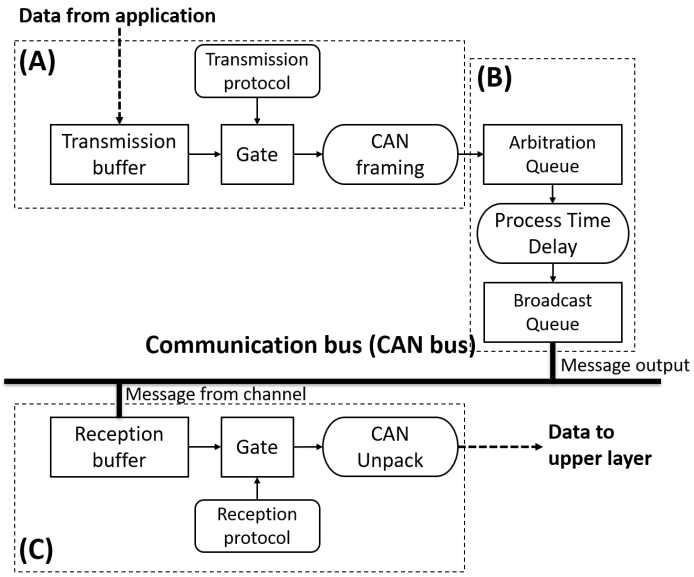

Figure 2. Block diagram for the channel implementation of the CAN bus. (A) transmission control block (B) medium access control and (C) reception control block.

C. Sensor Model Implementation

One of the main objectives of this work is to understand the effect of networked sensor communication in the variability of the measurement as perceived in the estimation algorithm. For that purpose, a model for the measurements was introduced in (5) for all the attitude sensors connected to the distributed communication bus as depicted in the AOCS Reference Architecture in Figure 1.

Figure 3, shows the block diagram for implementation of each sensor used in the simulation model.

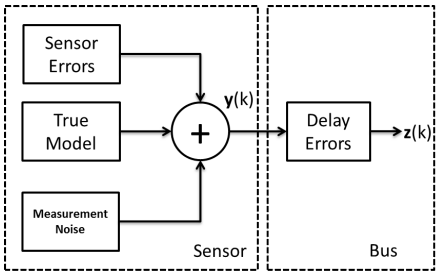

Figure 3. Sensors Block Diagram for Simulation Model

In that diagram, the true model propagator is in charge of generating the true value for sun sensors, magnetometers and gyroscopes using the spacecraft dynamics and kinematics equations. These measurements included sensor errors such as bias, drift and noise. The measurements are then supplied to the CAN controller for its transportation through the communication bus, where delays can cause additive errors.

D. Delay Error Modeling

The delay effect is not accounted within the sensor model, but it is modeled as a bus effect, as depicted in Figure 3. The delay of measurements is assumed produce an additive error that can be statistically estimated as follows

$$\mathbf{n}_k = \begin{cases} 0 & \text{Nominal Region,} \\ \boldsymbol{Norm}(\boldsymbol{\mu}_S, \boldsymbol{R}_S) & \text{Saturated Region} \end{cases} \quad (12)$$

Under nominal conditions, the effect of the delay errors is expected to be negligible, whereas when the communication channel reaches a saturation point, the delay is modeled as a normal distribution centered to a mean value for each sensor $\boldsymbol{\mu}_S$ with a measurement variance of $\boldsymbol{R}_S$. The representation of the measurement delay error as a random variable makes it possible to add it to the measurement model of the Kalman Filter used in the attitude estimation model introduced in (7).

E. Traffic Injection Model

To characterize the performance of the communication bus, it was necessary to generate additional traffic to emulate different satellite operation loads. The traffic injection function produces a burst which follows a Poisson Process with an inter arrival time (IAT) that is function of an average message arrival rate λ as described by the expression

$$IAT\ [s] = \begin{cases} \infty & \text{Nominal,} \\ \lambda e^{\lambda t} & \text{Additional Injected Traffic} \end{cases} \quad (13)$$

where, the additional injected traffic is combined with the nominal traffic in the communication bus that is exchanged between satellite sensors and subsystems. When the IAT is too large, there is no additional traffic injected in the communication bus.

IV. Simulation Experiments

This section is divided into two parts to illustrate the potential use of this communication model to represent satellite operation scenarios and describing on the experimental configuration. The simulation experiments were carried out on $MATLAB^{TM}$ Simulink 2016.

A. Satellites Operation Scenarios

Two case studies were considered in this paper. One involving an increase in the communication activities over the communication bus due to telemetry download procedures, and the second related to traffic increase in the bus due to additional data generated by the Attitude Determination and Control Subsystem (ADCS) sensors onboard of satellites with optical downlink communication. The primary objective of this section is to propose practical cases to verify and validate the communication bus implemented at simulation level.

1) Telemetry Files Download: The Delfi-Next satellite was launched in 2013. It was the second in a series of small satellite projects by Delft University of Technology (TU Delft) in the Netherlands. Delfi-Next was a triple unit CubeSat with a size of 10 cm x 10 cm x 34 cm. It had an active attitude determination and control subsystem, and a high-speed S-band transmitter communicating over a distributed linear bus with the rest of the spacecraft subsystems [33].

Delfi-Next satellite collected more than 300 telemetry parameters every two seconds. These parameters were broadcasted to the amateur radio network in the ground where they were collected and stored into a database for further analysis. Also, the telemetry data was stored locally in a database implemented in the onboard computer. The objective was downloading these files to the ground station in Delft by using the S-band transceiver onboard. The scenario created for this study consists in simulating the telemetry file download during one pass of the satellite over the ground station at least one time per day. The Delfi-Next telemetry database contained information from 316 parameters that were mapped into 50 CAN messages for its transmission from the OBC to the transceiver over the communication bus.

The telemetry files were expected to be downloaded in 20 seconds while the satellite passed over the ground station in Delft. The datarate of the communication bus in the satellite required to be configured at 1000 kbps to satisfy the download time. The arrival rate at the S-band transceiver was modeled to vary from 50 - 20000 CAN messages per second to describe a wide range of telemetry file sizes (up to 5MB).

That traffic was injected as additional traffic into the communication bus. That created different saturation levels in the bus of this spacecraft. That was considered for designing the experiments with the communication bus model described by (9).

2) ADCS Tracking for Optical Communication: Optical communication payloads are very sensitive to satellite pointing accuracy. For these satellites, its ADCS requires the implementation of both coarse and fine pointing algorithms to improve optical communication performance. In [34], the ADCS subsystem was enhanced by adding an optical beacon detector to provide on-line tracking adjustment and calibration to the optical transmitter with fast-steering mirrors.

In this simulation case, additional traffic generated by the beacon detector is injected to the communication bus to assess its impact on the estimation algorithm onboard the ADCS computer. The data rate of the communication bus is assumed fixed at 500 kbps. For the coarse pointing the extended Kalman filter fuses data from the ADCS sensors at 2 Hz, while for the fine pointing mode, the beacon detector and the centroid algorithm are working at 10 Hz using the same bus to communicate with all the other subsystems and components. The additional injected traffic was defined to vary in the range of 50 - 10000 CAN messages per communication cycle.

B. Experiment configuration

The spacecraft implementation consisted of a CAN network with up to 16 nodes including a 9-axis integrated inertial measurement unit with sun sensors, gyroscopes, magnetometers. Also included three reaction wheels, an optical beacon detector, an ADCS computer, a Command and Data Handling Subsystem (CDHS) computer, and a downlink/uplink communication module.

The simulation environment was set up based on the requirements from the case studies presented above. The experiments were intended to quantify the effect of design parameters and variables in the performance of the communication bus. The bus performance was related to the satellite state estimator performance, by quantifying the impact of bus saturation in the measurement error.

The measurement error was then determined indirectly by monitoring the sensors variance change at the ADCS computer. The experiment's hypothesis was that delayed measurements leaded to unstable values in the normalized variance of the sensors communicating over the satellite's bus. The additionally injected traffic was used to represent the dynamic behavior of the communication bus during telemetry download and ADCS tracking described in the case studies sections.

The experimental setup was chosen based on two conditions. Firstly, the bus configuration for data rate D_R and synchronization period T_{Sync}. The chosen values for these parameter were selected based on technology specifications for Commercial Off-The-Shelf transceivers for CAN. Secondly, the sensor sampling period T_{SS} and the network size N were selected based on previous ADCS configurations for CubeSats. The input parameters controlled during the simulation experiments were:

- Number of nodes in the network (fixed): 16
- CAN Message Length (fixed): 113 bits
- Transmission Queue size (fixed): 3 CAN messages.
- Reception Queue size (fixed): 3 CAN messages.
- Sensor sampling period T_{SS} (variable): 0.1 s and 0.5 s
- Channel synchronization period T_{Sync} (variable): 0.05 s and 0.01 s.
- Channel data rate D_R (variable): 500 kbps (ADCS Tracking case) and 1000 kbps (Telemetry Download case)
- Additionally Injected Traffic (variable): Up to 10000 msg/s for ADCS tracking and up to 20000 msg/s for telemetry download.

The experiment followed a 2^k full factorial design. The experimental execution was randomized to avoid the impact of confusion effects on the results. The output parameters collected for both case studies were:

- Mean communication bus utilization [%]
- Maximum communication bus utilization [%]
- Mean sensor measurement delay [s]
- Maximum sensor measurement delay [s]
- Measurement variance at source
- Measurement variance at AOCS

The bus utilization and sensor measurement delay were used to characterize the impact of injecting additional traffic into the communication bus, whereas the measurement variance was used as an indicator for determining the impact of delays in the AOCS estimation performance.

V. Results and Analysis

This section is divided into five subsections to present and discuss the findings of the experiments carried out. It also proposes an algorithm to mitigate saturation effects on the communication bus. The algorithm proposes a dynamic configuration of D_R and T_{Sync} parameters during satellite operations.

A. Bus Utilization

The bus utilization BU was obtained for both case studies as a function of the additionally injected traffic (AIT), the data rate (D_R), and the bus synchronization period (T_{Sync}). The goal of this experiment was reproducing different operation scenarios to characterize BU under heavy workloads of the satellite. In Figure 4 and Figure 5, the utilization for the CAN bus in the ADCS tracking case at 500 kbps, and the telemetry download case at 1000 kbps are respectively shown. The theoretical curves calculated from (9) were also included in the bus utilization profiles for comparison purposes.

Both bus utilization curves were marked with labels A, B and C to highlight specific behaviors on the bus. Firstly, there is the segmentation of the curve into two operation regions namely nominal and saturated. The limit between the nominal and saturated region was defined by the end of the linear demeanor in the bus utilization profiles (point B). The boundary between the nominal and saturated region was marked by the maximum physical capacity of the bus (point

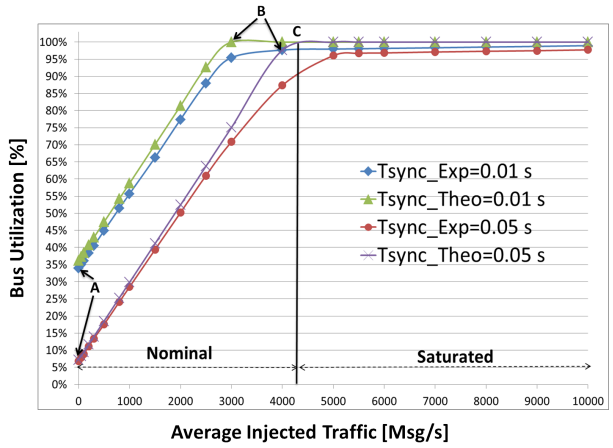

Figure 4. Bus utilization for the ADCS tracking Case Study at 500 kbps with T_{Sync} values of 0.01 s and 0.05 s obtained using both the analytical (Theo) and the simulated (Exp) models for a CAN network consisting of 16 nodes

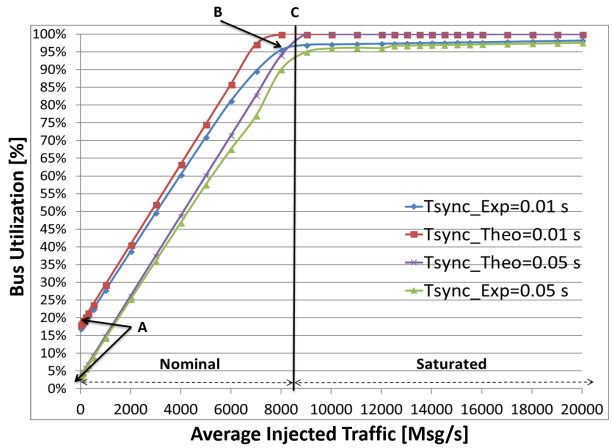

Figure 5. Bus utilization for the Telemetry Download Case Study at 1000 kbps with T_{Sync} values of 0.01 s and 0.05 s obtained using both the analytical (Theo) and the simulated (Exp) models for a CAN network consisting of 16 nodes

C), that can be determined analytically for each T_{Sync} and D_R combination. It is important to note that the values obtained experimentally with the simulation model, were consistent with the ones calculated with the analytical model.

The maximum physical capacity was calculated dividing the channel's data rate by the number of bits per CAN message. In the simulation model, each CAN message contained 113 bits, and it was kept as a constant parameter along the case studies. Therefore, the maximum physical capacity for 500 kbps and 1000 kbps were determined to be 4105 and 8530 messages per second at $T_{Sync} = 0.05$ s, respectively. Beyond the physical limit, the bus controller only allows the higher priority messages to be transmitted, while the lower priority components are held to access the communication bus causing extra delay, or even data loss for such data packets.

The bus saturation points were influenced by both data rate and channel synchronization period, as described by (9). The larger the value for these parameters, the bigger the amount of additional injected traffic that can be handled by the communication bus before reaching saturation conditions mentioned above.

From an implementation perspective, the values for data rate were determined by the technology used in the implementation of the CAN controller and transceiver, while the range of synchronization period was established according on the network size N and the sampling period for the sensors T_{SS} in the spacecraft. For practical cases, it is recommended to use a $T_{SS} \geq 10T_{Sync}$ to avoid aliasing effects of the communication channel in the sensor measurements [35].

The third aspect that was analyzed in the bus utilization profiles was the starting point of the curve in the nominal region (Point A) for different synchronization periods T_{Sync}. A sensitivity analysis was conducted to understand the effect of the number of nodes connected to the bus, and the T_{Sync} parameter on the initial value for the bus utilization. Figure 6 shows the results of this analysis for a reference data rate of 1000 kbps, a sensor sampling rate of 0.5 s and four CAN synchronization periods: 0.001 s, 0.01 s, 0.05 s, and 0.1 s.

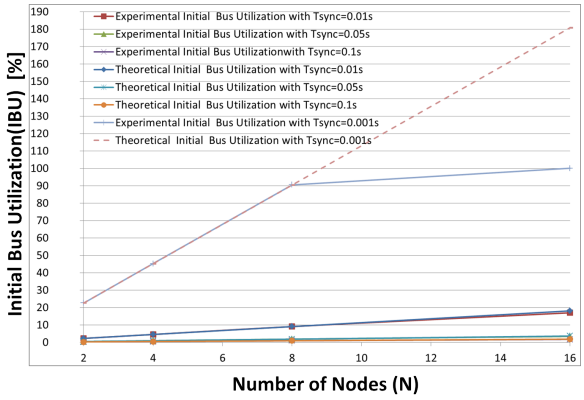

Figure 6. Effect of Network Scale in the Initial Bus Utilization for T_{Sync}= 0.001 s, 0.01 s, 0.05 s, and 0.1 s

It is important to note that for all $T_{Sync} \geq 0.01\ s$ the bus utilization varied linearly with the number of nodes as described by (9), as long as the bus does not reach the maximum capacity, meaning that it was operating on the nominal region. For the sensitivity analysis, no additional traffic is injected ($\lambda \rightarrow \infty$), and a fixed data rate of 1000 kbps was used, so that the bus utilization was a function only of the number of nodes in the network N and the synchronization period T_{Sync}. Figure 6 shows consistency between the theoretical and the experimental model used in the simulation case studies. It also showed that by varying T_{Sync} the slope of the saturation curve can be controlled, which can be used to mitigate by design the impact of additional traffic injected into the bus.

B. Measurements Delay

During the experiments, both the mean and the maximum delay for sensors communicating over the spacecraft bus were monitored and recorded.

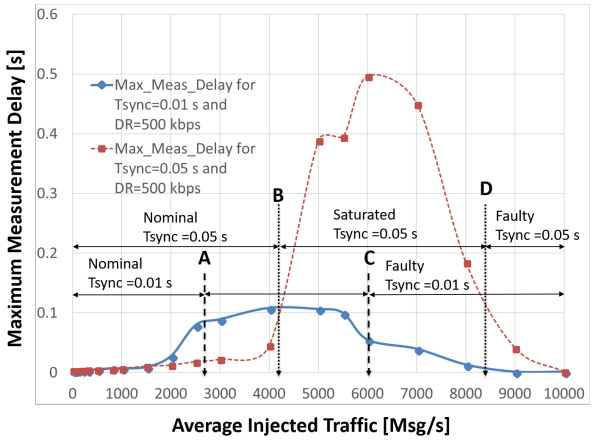

Figure 7. Maximum Delay observed as a function of the Additional Traffic Injected for a Communication Bus working at D_R=500 kbps, with T_{Sync}= 0.01 and 0.05 s, respectively

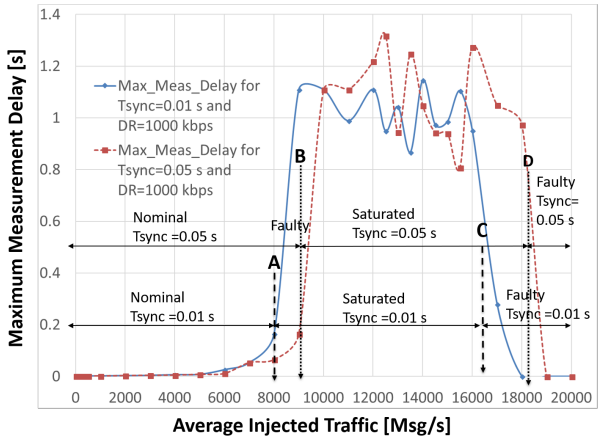

Figure 8. Maximum Delay observed as a function of the Additional Traffic Injected for a Communication Bus working at D_R=1000 kbps, with T_{Sync}= 0.01 and 0.05 s, respectively

Figure 7 and Figure 8 show the effect of channel saturation in the maximum delay observed for both data rates 500 kbps and 1000 kbps, respectively. Each plot shows the delay recorded for T_{Sync}=0.01 s and T_{Sync}=0.05 s as a function of the injected traffic on the communication bus. In both Figures, labels A and B define the saturation point for each T_{Sync} configuration. Label A defines the saturation point for T_{Sync}=0.01 s and label B defines the saturation point for T_{Sync}=0.05 s. It is important to note that for both data rates, when T_{Sync} was increased, also the capacity of the bus to deal higher injected traffic without increasing the delay significantly was incremented. This behavior is consistent with the bus utilization model presented by (9). However, it was observed that the higher the T_{Sync}, also the bigger the maximum delay observed on the saturated region for both data rates. For design purposes, the maximum T_{Sync} possible is limited by the sensor sampling period T_{SS} requirements as discussed above.

The data rate also affected the maximum delay observed during the experiments. The higher the data rate D_R, the bigger the maximum delay value observed. Labels C and D were used to determine the traffic level under which the simulation model fails. Beyond that point, the delay was observed to fall abruptly to zero. The saturated region in both plots showed that the simulation model can handle approximately the double of the theoretical channel capacity before failing.

The oscillating behavior observed on the delay curves in the saturation region is due to the way in which the additionally injected traffic was implemented. It used a random function, together with λ to generate the exponential inter-arrival times for the injected messages as described by (13).

The effect of additional injected traffic and synchronization period T_{Sync} was compared statistically. After performing a two-sample t-test for the nominal operation region, there was no statistical significance for either the data rate and the synchronization period. That means that while the bus operates in the nominal region, the communication bus was not affected by the extra traffic injected on the communication bus.

That is not the case for the saturated region, where both mean, and variance of the delay are statistically significant. There was an effect of saturation in the mean and variance of the delay for the both case studies. The effect of synchronization period showed a difference in the p-value during the test.

C. Effect of Delays in Measurements Variance

So far, the effect of data rate and synchronization period for handling heavy intra-communication workloads on the communication bus has been shown and discussed. It was demonstrated that saturation of the communication channel generated a significant delay in the arrival of measurements to the ADCS computer. From the attitude estimation perspective, it is necessary to understand the effect of these delays in the quality of measurements provided to the ADCS computer in charge of executing the estimation algorithms described by 7. For that purpose, a series of experiments were carried out to quantify the impact of delays in the variance of measurements received at the ADCS computer for both satellite operations scenarios.

The variance performance of the sensors measurements was monitored and recorded both in the sensors and the ADCS computer during a pass of the satellite over the ground station with the intention of quantifying the effects of delay. Then, it was normalized for comparison purposes. After that, the change in the normalized measurements variance was plotted as a function of the injected traffic on the communication bus. The measurements time stamp was collected both at the source in the sensor, as well as, in the destination of the ADCS computer for both case studies. Finally, a MATLAB script calculated delays in the data propagation within the communication bus.

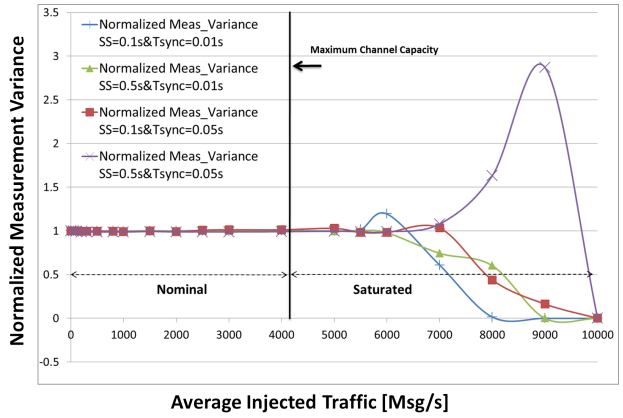

Figure 9. Measurements Variance Profile for a Communication Bus Operating at D_R=500 kbps with T_{Sync}= 0.01 and 0.05 s, and Sensor Sampling Periods SS= 0.1 s and 0.5 s

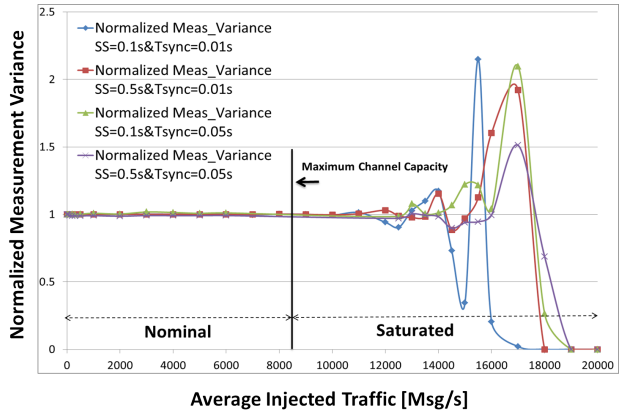

Figure 10. Measurements Variance Profile for a Communication Bus Operating at D_R=1000 kbps with T_{Sync}= 0.01 and 0.05 s, and Sensor Sampling Periods SS= 0.1 s and 0.5 s

Figure 9 and Figure 10 show the normalized measurements variance profiles for two sensor sampling periods SS (0.1 s and 0.5 s), two bus synchronization periods T_{Sync} (0.01 s and 0.05 s), and two data rates D_R (500 kbps and 1000 kbps). One can notice that in the nominal region the variability kept constant to the unit value, meaning that the additionally injected traffic did not affect the measurements quality, and therefore the state estimation algorithm can neglect the effects of delay on its measurements model. That was not the case for the saturated region, where the additionally injected traffic increased the variance of measurements received at the ADCS computer. After the saturation of the communication bus, the variance started to exhibit an oscillating behavior for the different bus configurations. From the analysis of these profiles, it was clear that the change in variance was more sensitive to the datarate D_R rather than the synchronization period T_{Sync}, which is consistent with the maximum delay observed.

In Figure 10 the oscillation characteristic for 1000 kbps appeared later than for the 500 kbps data rate in Figure 9. It is also clear that this variance demeanor increased with the amount of injected traffic in the bus, due to the delay in the arrival of measurement samples to the ADCS computer.

A statistical analysis of the measurements variance for the communication bus operating under different configuration values for D_R and T_{Sync} was executed. As result of these analysis, there is a noticeable difference in the measurement variance deviation for the saturated region compared to the nominal region. After performing a 2-sample-t-test for both cases there is no significance for the means, but there is for the variance of the measurements. Again, a higher sensibility is linked to the data rate than synchronization period, while there is no effect coming from the sensor sampling period SS.

Table I summarizes the effect of additional injected traffic in the quality of measurements received at the ADCS computer for a communication bus operating in both nominal (AIT=0 msg/s) and saturated region (6000 and 12000 msg/s for 500 and 1000 kbps, respectively). It shows again that the effect of increasing data rate was bigger than the effect of increasing T_{Sync} on the measurements variance of a saturated bus. The same behavior was also observed during the maximum delay characterization in Section V-B.

Table I. VARIANCE INCREASE OF MEASUREMENTS AT THE ADCS COMPUTER FOR NOMINAL AND SATURATED COMMUNICATION BUSES UNDER DIFFERENT DATA RATES AND T_{Sync} VALUES

Data rate [kbps]	T_Sync [s]	Variance Increase for the Bus Operating in the Nominal Region	Variance Increase for the Bus Operating in the Saturated Region
500	0.01	0.1%	1%
	0.05	2%	3.5%
1000	0.01	0.05%	18%
	0.05	2%	10%

D. Mitigation Strategy for Delay in Saturated Buses

The experimental results have shown a direct relationship between bus saturation, measurements delay and measurements variance. However, for operational purposes, it is also interesting to include mechanisms to deal with bus saturation once the satellite has been deployed. Experimental results have also shown a direct link between the power consumption in the CAN transceiver/controller and the bus utilization as presented by [36].

Figure 11 shows an algorithm proposed for saturation detection and recovery using an agent-based approach. The agent assumes a CAN controller with flexible data rate capabilities, running a synchronous implementation of CAN protocol, for instance, CAN-open. The algorithm balances both the data rate D_R and the synchronization period T_{Sync} to compensate the additional injected traffic, and satisfying the system budgets.

VI. CONCLUSIONS AND OUTLOOK

This paper has introduced a simulation model to assess the effect of increasing the traffic volume over a distributed communication bus of a spacecraft. Experiments showed that the effect of delays for distributed communication buses operating in the nominal region, do not require modifying

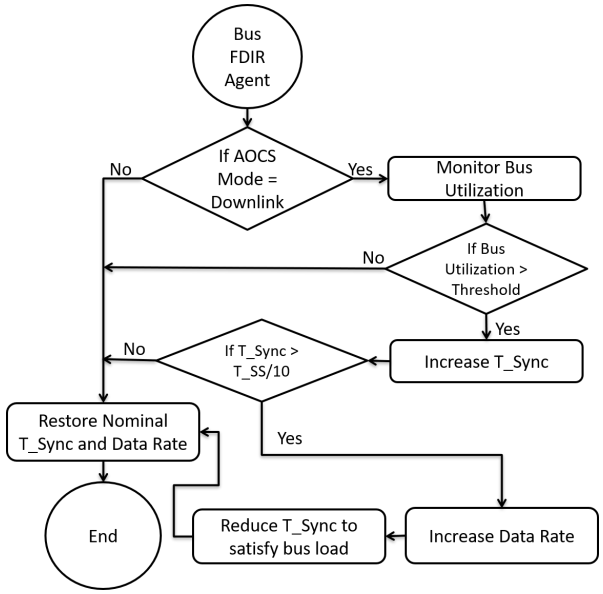

Figure 11. Flowchart for Saturation Detection and Recovery Algorithm Proposed for a Distributed Communication Bus

the measurement model in the estimation algorithm. The bus utilization in the nominal region showed a linear behavior as a function of the additionally injected traffic. The bus utilization also showed a linear dependency with the network size. The results from the simulation model show consistency with the bus utilization expression proposed in (9).

There is also evidence of statistical significance for data delay in the saturated region compared to the nominal region, leading to a degradation of sensor measurement variability of up to 18%. The experiments also showed an increase in the measurements variability for a saturated bus compared to nominal conditions. Controlling bus utilization can be used as an operational parameter to mitigate performance issues in ADCS due increased information traffic.

Future research will consider the implementation of the Algorithm in Figure 11 as well as a comparison of the results of this implementation with other communication protocols such as I2C to verify and validate the findings. It is essential to consider the newer capabilities of CAN protocols such as flexible data rate for further implementations. The computational burden will be assessed in future experiments to understand its effects. Additionally, the presented algorithm can be assessed to improve other network-based industrial processes such as Programmable Logic Controllers (PLC) networks, or distributed systems in automotive applications. Finally, a sensitivity analysis shall be conducted to explain the effects of channel configuration on the maximum delay observed on the numerical model.

REFERENCES

[1] F. Moll, H. Weinfurter, M. Rau, C. Schmidt, G. Melén, T. Vogl, S. Nauerth, and C. Fuchs, "Aerospace laser communications technology as enabler for worldwide quantum key distribution," *Proc. SPIE 9900*, pp. 99 000K–1, 2016.

[2] M. Toyoshima, "Trends in satellite communications and the role of optical free-space communications," *J. Opt. Netw.*, vol. 4, no. 6, pp. 300–311, Jun 2005. [Online]. Available: http://jon.osa.org/abstract.cfm?URI=jon-4-6-300

[3] Y. Cheng and T. G. Robertazzi, "Distributed computation with communication delay (distributed intelligent sensor networks)," *IEEE transactions on aerospace and electronic systems*, vol. 24, no. 6, pp. 700–712, 1988.

[4] Y.-C. Cheng and T. G. Robertazzi, "Distributed computation for a tree network with communication delays," *IEEE transactions on aerospace and electronic systems*, vol. 26, no. 3, pp. 511–516, 1990.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[6] S. Li, H. Du, and P. Shi, "Distributed attitude control for multiple spacecraft with communication delays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 1765–1773, 2014.

[7] D. Morris, "Command and control systems," in *Communication for Command and Control Systems*, ser. International Series on Systems and Control, D. Morris, Ed. Pergamon, 1983, pp. 3 – 50.

[8] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systemsa survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[9] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control engineering practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[10] T. Wang, J. Qiu, H. Gao, and C. Wang, "Network-based fuzzy control for nonlinear industrial processes with predictive compensation strategy," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2137–2147, Aug 2017.

[11] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 98–101, 1986.

[12] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," *Information Sciences*, vol. 380, pp. 117 – 131, 2017.

[13] D. J. Morris, *Communication for Command and Control Systems: International Series on Systems and Control*. Elsevier, 2014.

[14] D. Paret, *Multiplexed networks for embedded systems: CAN, LIN, Flexray, Safe-by-Wire...* John Wiley & Sons, 2007.

[15] W. Lawrenz, "Can system engineering," *From theory to practical applications, New York*, 1997.

[16] S. Ning, C. Nan, Z. Bingjun, P. Dawei, and Z. Guohua, "Effects of can communication delay on abs fuzzy control [j]," *Journal of Jiangsu University (Natural Science Edition)*, vol. 4, p. 008, 2010.

[17] D. Gwaltney and J. Briscoe, "Comparison of communication architectures for spacecraft modular avionics systems," *Tech. Rep. NASA/TM-2006-214431*, 2006.

[18] H. L. Alexander, "State estimation for distributed systems with sensing delay," in *Orlando'91, Orlando, FL*. International Society for Optics and Photonics, 1991, pp. 103–111.

[19] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: exact solution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 769–777, 2002.

[20] F. Daum, "Nonlinear filters: beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[21] K. You, N. Xiao, and L. Xie, *Kalman Filtering with Faded Measurements*. London: Springer London, 2015, pp. 223–237.

[22] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[23] A. W. Marshall and I. Olkin, "A family of bivariate distributions generated by the bivariate bernoulli distribution," *Journal of the American Statistical Association*, vol. 80, no. 390, pp. 332–338, 1985.

[24] Y. Zhang and J. Jiang, "Fault tolerant control system design with explicit consideration of performance degradation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 838–848, 2003.

[25] I. Broster and A. Burns, "Timely use of the can protocol in critical hard real-time systems with faults," in *Real-Time Systems, 13th Euromicro Conference on, 2001.* IEEE, 2001, pp. 95–102.

[26] J. Bouwmeester, G. Brouwer, E. Gill, G. Monna, and J. Rotteveel, "Design status of the delfi-next nanosatellite project," in *61st International Astronautical Congress, Prague, Czech Republic, 27 September-1 October 2010*. International Astronautical Federation, 2010.

[27] H. Kimm and M. Jarrell, "Controller area network for fault tolerant small satellite system design," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*. IEEE, 2014, pp. 81–86.

[28] M. Khurram and S. M. Y. Zaidi, "Can as a spacecraft communication bus in leo satellite mission," in *Proceedings of 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST 2005.*, June 2005, pp. 432–437.

[29] L. Almeida, P. Pedreiras, and J. A. G. Fonseca, "The ftt-can protocol: Why and how," *IEEE transactions on industrial electronics*, vol. 49, no. 6, pp. 1189–1201, 2002.

[30] T. A. Johansen, I. Petersen, J. Kalkkuhl, and J. Ludemann, "Gain-scheduled wheel slip control in automotive brake systems," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 799–811, Nov 2003.

[31] I. Petersen, "Wheel slip control in abs brakes using gain scheduled optimal control with constraints," *Department of Engineering Cybernetics. Norwegian University of Science and Technology, Trondheim, Norway*, 2003.

[32] K. W. Tindell, H. Hansson, and A. J. Wellings, "Analysing real-time communications: controller area network (can)," in *1994 Proceedings Real-Time Systems Symposium*, Dec 1994, pp. 259–263.

[33] J. Guo, J. Bouwmeester, and E. Gill, "In-orbit results of Delfi-n3Xt: Lessons learned and move forward," *Acta Astronautica*, vol. 121, pp. 39–50, Apr. 2016.

[34] T. Nguyen, K. Riesing, R. Kingsbury, and K. Cahoy, "Development of a pointing, acquisition, and tracking system for a cubesat optical communication module," in *Proceedings of SPIE–the Society of Photo-Optical Instrumentation Engineers*. SPIE, 2015.

[35] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*. Addison-Wesley Reading, MA, 1994, vol. 3.

[36] E. G. A. Orsel, "Power modelling and optimization of a communication bus for small satellite missions," Master's thesis, Delft University of Technology, the Netherlands, 2016.

Johan Carvajal-Godinez has an Electronic Engineering degree with a Master in Modern Manufacturing Systems from the Costa Rica Institute of Technology. He is currently a Ph.D. Researcher in the Space Engineering Department at the Faculty of Aerospace Engineering of the Delft University of Technology.

Jian Guo has his Ph.D. degree from University of Leeds in UK and M.Sc. and B.Sc. from Northwestern Polytechnical University in China. He is currently an Assistant Professor in the Space Engineering Department at the Faculty of Aerospace Engineering of the Delft University of Technology. He is also a Theme Leader of TU Delft Space Institute.

Eberhard Gill received a diploma in physics and holds a PhD in theoretical astrophysics of the Eberhard-Karls-University Tuebingen, Germany. He holds a Master of Space Systems Engineering of the Delft University of Technology. He is currently the Department Head of the Space Engineering Department at the Faculty of Aerospace Engineering of the Delft University of Technology. He is also the Director of TU Delft Space Institute.