Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

1 kHz Remote Control of a Balancing Robot with Wi-Fi–in–the–Loop ⋆

Francesco Branz, [∗,1] **Riccardo Antonello,** [∗]
Matthias Pezzutto, [∗] **Federico Tramarin,** [∗∗] **Luca Schenato** [∗]

[∗] *Department of Information Engineering, University of Padova, Padova, Italy*
[∗∗] *Department of Management and Engineering, University of Padova, Vicenza, Italy*

Abstract:
Countless industrial applications can potentially benefit from the implementation of wireless control systems, leading to a widespread research effort to investigate new solutions in the field. Nevertheless, currently available wireless communication standards for industrial automation are not able to achieve high control frequencies. In particular, time–critical applications (e.g. industrial robotics and manipulation) require high sampling frequencies to be properly implemented. The higher throughput provided by IEEE 802.11 (Wi-Fi) can theoretically tame critical applications, although reliability is a key issue. In this work Wi-Fi is adopted to increase the achievable control rates up to 1 kHz, while reliability is guaranteed by mitigating communication flaws through model–based estimation techniques. The core of the proposed approach relies on a modified Kalman filter that exploits a buffer of incoming measures to account for delayed data packets. The proposed solution is validated through a hardware–in–the–loop experiment that features actual Wi-Fi hardware and a commercial embedded PC board. The obtained results give a preliminary, yet valuable, validation of the proposed approach testing the solution on relevant hardware.

Keywords: Control over networks; Control under communication constraints; Networked embedded control systems; Control and estimation with data loss; Hardware–in–the–loop simulation.

1. INTRODUCTION

Networked Control Systems (NCSs) have been thoroughly investigated over the last decades and limitations due to communication flaws (i.e. packet losses and random delays) have been dealt with a wide range of solutions as reported by Xia et al. (2015) and Zhang et al. (2016). Nevertheless, industrial use of wireless networks for control purposes is still uncommon and several implementation aspects have not been explored, yet. Although a large body of literature on NCSs describes the key issues, practical approaches and procedures to connect a plant and a controller over wireless are missing. Rigorous and verified solutions to translate theory into practice are still to be developed.

To this end, the authors consider a NCSs structure that features a physical plant equipped with an *embedded system*, consisting of a computational unit whose tasks are to control and to autonomously manage the plant. The embedded system is equipped with a *wireless network interface* that is used to connect the plant to a remote stabilizing controller, to a supervisory unit, or to other plants, meeting the flexibility and modularity demands of modern applications. Many different WLAN standards are available to implement control applications over wireless. However, physical and MAC layers of existing standards for automation purpose (WirelessHart, ISA100.11a, 6TiSCH, WISA) are conceived to serve low–frequency applications maximizing reliability, with the drawback of being unable to support high sampling rates. As noted by Petersen and Carlsen (2011), their data–rate and time–slotted channel prevent sampling time to be reduced below 50 ms (see also Zhu et al. (2012)). The IEEE 802.11 standard (i.e. Wi-Fi) presents much higher data–rates and can theoretically break the sampling frequency boundary, aiming even at challenging scenarios with control frequencies on the order of 1 kHz (e.g. multi–agent robotic manipulation). Furthermore, Wi-Fi is attractive for its simplicity, availability and compatibility with existing hardware. On the other hand, the higher data–rates come at the cost of an increased unreliability that needs to be carefully evaluated.

In Branz et al. (2019b), the authors connected the sensor and the control unit through a Wi-Fi network (feedback path), while the controller and the actuator (forward path) were connected by a standard quasi–ideal point–to–point link, practically realized through an Ethernet connection. The proposed solution was the adoption of the optimal estimator described by Schenato (2008) integrated with the optimal infinite–horizon regulator. The resulting control algorithm consists of a modified Kalman filter with a finite–length buffer and a static–gain state feedback. Experimental results show relevant improvements in robustness and effectiveness in the case of disturbed communication channel.

This work generalizes the problem a little further by considering a wireless connection for both forward and feedback paths. The new case is much more challenging due to the information asymmetry between the plant and the control unit regarding the sequence of inputs and the state of the system (see Imer et al. (2006)). A straightforward solution for low–frequency applications is to achieve reliable data transmission by using Transmission Control Protocol

⋆ This work is partially supported by University of Padova, project MAgIC SCHE_SID17_01.
[1] Corresponding author: F.B. francesco.branz@unipd.it.

(TCP) at the application layer, thus removing the asymmetry between the two sides of the control architecture. Drawbacks of TCP–like solutions include (1) a suitable acknowledgement mechanism is needed causing increased latency, (2) a larger number of transmitted packets at the physical layer (at least 50% higher) demanding higher bandwidth and computational resources, (3) the possibility to randomly lose the transmission acknowledgment making the LQG control technically difficult (see Garone et al. (2008)). These aspects make TCP unsuitable for the proposed architecture, particularly when high–frequency applications are implemented on embedded systems as assessed in Branz et al. (2019a). User Datagram Protocol (UDP) is selected for the lightweight structure and for the easier implementation of communication routines (at the application layer), at the price of a degraded reliability that affects the control performance. As shown by Schenato et al. (2007), separation principle does not hold over lossy links under UDP–like protocol, even in the linear case. In particular, the control input is a nonlinear function of the state estimate and the optimal estimate is a non linear function of the control input. The optimal steady–state linear controller and the optimal steady–state estimator without acknowledgement are provided by Sinopoli et al. (2008). Two possible strategies to address missing packets at the actuator are studied by Schenato (2009): the zero–input strategy, where the input is set to zero in case of packet loss, and the hold-input strategy, where the last valid control input is kept. None of the two proves better than the other and their behavior is strongly dependent on system and channel conditions. A practical approach is the adoption of suboptimal algorithms that represent an effective trade–off between the performances and the computational load. Relevant works include Lin et al. (2015) that proposes optimal and suboptimal filters, and Lin et al. (2017) that introduces the optimal control in the case of a smart sensor able to transmit a local estimate.

In this work the authors investigate the use of Wi-Fi in combination with a modified Kalman filter that takes care of communication flaws, implementing only minor modifications to the communication policy. The contribution of this work is the implementation of the mentioned approach to tame two–way wireless control on embedded systems and its validation through a hardware–in–the–loop experiment that involves commercial Wi-Fi hardware.

2. METHOD

The problem considered in this paper is to reliably operate a closed–loop control system over Wi-Fi, possibly coping with the intrinsic limitations of the IEEE 802.11 standard in guaranteeing a timely and reliable data delivery. On one hand, the proposed method consists to select a protocol that assures low latency delivery (i.e. UDP), to tune the number of transmission retries of the wireless interface, and to carefully implement time–efficient receiver routines. On the other hand, a modified Kalman estimator is introduced at the controller side, with the aim of mitigating, through model-based state prediction, possible measurement feedback interruptions due to packet losses on the Wi-Fi link. This is in contrast with the conventional emulation–based approach (see Walsh et al. (2002); Nešić and Teel (2004); Maass et al. (2017)). No modifications are envisaged for the Wi-Fi standard, so that the proposed architecture can be readily implemented by resorting to off–the–shelf components.

A simplified block diagram of the wireless NCS architecture considered in this paper is reported in Fig. 1. It consists of a closed–loop control system where both the control commands (forward path) and the plant output

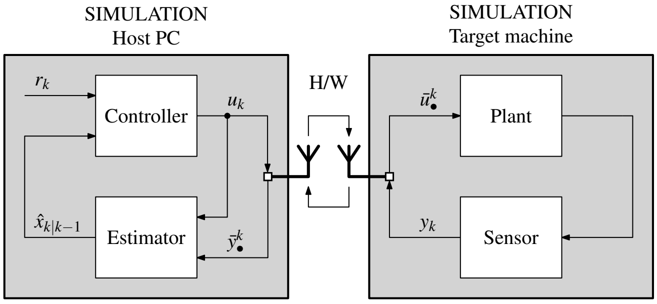

Fig. 1. Wireless NCS system architecture.

measurements (feedback path) are exchanged over a Wi-Fi channel. The controller is running on a dedicated unit not embedded with the plant. Let

$$\begin{cases} x_{k+1} = Ax_k + B\bar{u}^k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{1}$$

be a model of the plant, with $x_k,\ w_k \in \mathbb{R}^n$, $\bar{u}^k \in \mathbb{R}^m$, and $y_k,\ v_k \in \mathbb{R}^p$. Moreover, assume $w_k \sim \mathcal{N}(0,\ Q)$ with Q positive semidefinite, $v_k \sim \mathcal{N}(0,\ R)$ with R positive definite, and $x_0 \sim \mathcal{N}(\hat{x}_0,\ P_0)$ with P_0 positive semidefinite. At time instant k, a command $\bar{u}^k$ is applied to the plant, while the new command u_{k+1} and output measure y_k are packetized and transmitted, with an attached time-stamp, by either side of the link. The arrival processes for the two quantities are modelled with the two random variables:

$$\theta_t^k = \begin{cases} 1 \text{ if } u_t \text{ is available at actuator at } k \geqslant t \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

and

$$\gamma_t^k = \begin{cases} 1 \text{ if } y_t \text{ is available at estimator at } k \geqslant t \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

The variable θ_k^k is assumed to be a i.i.d. Bernulli random variable with expected value θ. On the other hand, no assumptions are made on γ_t^k, given the difficulties of obtaining a satisfactory characterization.

At the plant side, the control input $\bar{u}^k$ is selected according to the hold strategy (see Schenato (2009)):

$$\bar{u}^k = (1 - \theta_k^k)\,\bar{u}^{k-1} + \theta_k^k\, u_k \tag{4}$$

which corresponds to apply, at any instant, the latest received command value while discarding out-of-time packets. This choice is superior to the zero strategy when loss probability is low, i.e. when the channel condition is good.

For the computation of the command u_k, two alternative strategies can be adopted, namely the *emulation–based* and the *model–based* approach. In the former, the control design is carried out as if the communication between the controller and the plant is ideal, i.e. network non–idealities are neglected. This approach is reasonable provided that the dynamics of the plant to be controlled is sufficiently slower than the transmission rate of the network. Any design can be pursued (e.g. LQG); however, when implemented on the real NCS, the emulation–based controller has to face with a possible irregular measurement feedback due to communication issues like packet losses and delays. Typically, at any sampling instant it is assumed to feed the controller with the most updated measure, which implies to adopt a hold–strategy similar to (4), namely:

$$\bar{y}^k = (1 - \gamma_k^k)\,\bar{y}^{k-1} + \gamma_k^k\, y_k \tag{5}$$

Although rather simplistic, this approach is appreciated because the design (nothing more than classic control theory is required) and the implementation (e.g. no time stamps and no detection of packet loss) are straightforward. At the same time, it turns out to be satisfactory when losses are a few, and long delays are excluded. This is usually achieved by common wireless standards when systems are slow (1-50 Hz).

Differently from the emulation–based approach, in the model–based approach the control design is performed by accounting for the network non–idealities. A modified Kalman estimator with data buffering is employed to perform a model–based prediction whenever a measurement feedback is not available. The approach proposed in this paper is an extension of the design case that accounts for the measurement delays/drops, in which no reliable control input delivery is assumed. In a framework where

$$\mathcal{I}^k_{\mathrm{MB}} = \left\{\gamma^k_0, \cdots, \gamma^k_{k-1},\, \bar{y}^k_0, \cdots, \bar{y}^k_{k-1},\, u_0, \cdots, u_{k-1}\right\} \quad (6)$$

denotes the information set available to the controller at time k, with now

$$\bar{y}^k_t = \gamma^k_t y_t \quad (7)$$

being the new measurement model (different from (5)), it is known that separation principle does not hold, so that the optimal control and state estimator cannot be independently designed. Since the joint derivation may be cumbersome, in the following a suboptimal solution is proposed, where the estimator is designed first, and the controller is designed next by assuming full–state access.

The optimal estimator can be derived by following Schenato et al. (2007) and Schenato (2008) but considering the hold–control strategies. Let

$$\widehat{x}^{\mathrm{MB}}_{k|k-1} = \widehat{x}^k_{k|k-1} \quad (8)$$

denote the optimal state estimate given $\mathcal{I}^k_{\mathrm{MB}}$. The following iterative procedure provides a method for computing it:

$$\widehat{x}^k_{t|t-1} = \mathbb{E}\left[x_t|\mathcal{I}_k\right] = A\widehat{x}^k_{t-1|t-1} + B\mathbb{E}\left[\bar{u}^{t-1}|\mathcal{I}_k\right] \quad (9)$$

$$P^k_{t|t-1} = \mathbb{E}\left[(x_t - \widehat{x}^k_{t|t-1})(x_t - \widehat{x}^k_{t|t-1})|\mathcal{I}_k\right] \quad (10)$$

$$\widehat{x}^k_{t|t} = \widehat{x}^k_{t|t-1} + \gamma^k_t K_t(\bar{y}^k_t - C\widehat{x}^k_{t|t-1}) \quad (11)$$

$$P^k_{t|t} = P^k_{t|t-1} - \gamma^k_t P^k_{t|t-1}C'(CP^k_{t|t-1}C' + R)^{-1}CP^k_{t|t-1} \quad (12)$$

$$K_t = P^k_{t|t-1}C'(CP^k_{t|t-1}C' + R)^{-1} \quad (13)$$

starting from

$$\widehat{x}^k_{k-N|k-N-1} = \widehat{x}^{k-1}_{k-N|k-N-1},\;\; P^k_{k-N|k-N-1} = P^{k-1}_{k-N|k-N-1}$$

if $t > N$ or from

$$\widehat{x}^t_{0|-1} = \widehat{x}_0, \quad P^t_{0|-1} = P_0$$

otherwise, where N denotes the length of the buffer [2] that stores the newest data. For space limitation reasons, the closed–form expressions of $P^k_{t|t-1}$ and $\mathbb{E}\left[\bar{u}^{t-1}|\mathcal{I}_k\right]$ are omitted. The resulting estimator is a time–varying modified Kalman filter endowed with a buffer (see Fig. 2). Since $P^k_{t|t}$ depends on the particular arrival sequence, it can not be computed in advance and K_t does not converge to a steady–state gain.

At each time instant, the estimating procedure is iterated from the oldest measurement received at that instant: from such starting point, the iteration involves a prediction (open–loop) step if a measurement is not available in the buffer (i.e. $\gamma^k_t = 0$), or an estimate (closed–loop) step otherwise (i.e. $\gamma^k_t = 1$). The buffer is used to store all

[2] Indeed, the solution is optimal only if the buffer has an infinite length. However, for implementation purposes, a finite buffer length version is typically preferred, leading however to a sub-optimal solution (see Schenato (2008)).

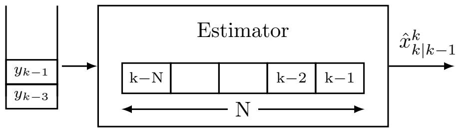

Fig. 2. Proposed estimator.

the relevant estimator quantities, i.e. the state prediction $\widehat{x}^k_{t|t-1}$ and estimate $\widehat{x}^k_{t|t}$, the covariance matrices $P^k_{t|t-1}$ and $P^t_{t|t}$ (of, respectively, the state prediction and estimation errors), the control inputs u_t, and the received plant outputs y_t, together with the arrival sequence γ^k_t. The buffer length N represents the maximum delay accounted by the estimator: any packet delay larger than N sampling periods will be interpreted as a packet loss. The sizing of the buffer length can be performed after measuring the average packet delay in the communication link, as described in Sec. 4.

In order to simplify the implementation, the actual error covariance $P^k_{t|t-1}$ are substituted with the equation for the case with reliable channel

$$\widehat{x}^k_{t|t-1} = \mathbb{E}\left[x_t|\mathcal{I}_k\right] = A\widehat{x}^k_{t-1|t-1} + Bu_{t-1} \quad (14)$$

$$P^k_{t|t-1} = AP^k_{t-1|t-1}A' + Q \quad (15)$$

The rationale behind this choice is that under hold strategy, especially when sequences of consecutive packet losses are small, the difference between the u_t and $\bar{u}_t$ is small in average, thus the above approximation is good.

Assuming that the state is completely accessible (when the corresponding packet is not lost), the optimal controller can be found by following Schenato (2009), namely

$$u^{\mathrm{MB}}_k = L\widehat{x}^{\mathrm{MB}}_{k|k-1} \quad (16)$$

where L and S satisfy the equations

$$\begin{bmatrix} S_{11} & S_{12} \\ S'_{12} & S_{22} \end{bmatrix} = W + A'S_{11}A - (1-\theta)A'(S_{11}B + S_{12})T^{-1}(B'S_{11} + S'_{12})A \quad (17)$$

$$T = U + S_{22} + B'S_{11}BS'_{12}B + B'S_{12} \quad (18)$$

$$L = -T^{-1}(B'S_{11} + S'_{12})A. \quad (19)$$

3. BENCHMACK APPLICATION

The approach proposed in Sec. 2 is tested on a control application consisting of the remote stabilization of a Segway–like vehicle (balancing robot), characterized by an unstable dynamics that requires continuous control adjustments to accomplish the task. It is considered a simplified, yet significant experiment to set a baseline for more complex consumer and industrial control applications. The reference prototype considered in this paper is illustrated in Fig. 6b. Two wheel speed servo–loops are implemented on the robot; they receive an acceleration equivalent command from an outer controller, hosted on an external platform (host PC). The same speed reference (obtained by integration of the acceleration command) is provided to the servo–loops. This implies that the robot moves along a straight line, and the dynamics is constrained on its sagittal plane. With this choice, it is possible to simplify the design, by simply considering a planar-equivalent model of the robot. Obviously, no lateral movements are commanded with this simplified configuration. The measurements sent to the controller are the wheel angular position δ (same for the two wheels), retrieved with dedicated on–board encoders, and the robot tilt angle ϑ, obtained by

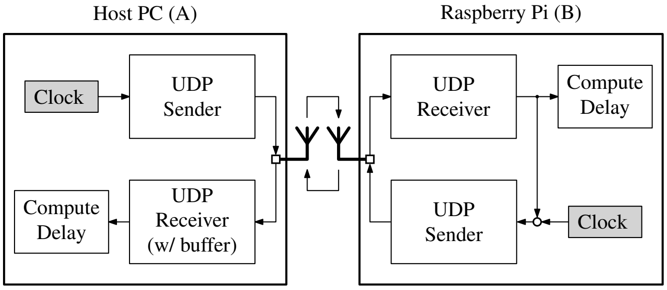

Fig. 3. Wi-Fi hardware–in–the–loop characterization experiment.

fusing the information (complementary filtering) provided by the onboard IMU sensor (accelerometer and gyroscope). Both the acceleration command and the measurements are exchanged over a Wi-Fi link.

The mathematical model required for the controller and state estimator design has the form:

$$\alpha_2\ddot{\vartheta} + \alpha_1\dot{\vartheta} + \alpha_0\vartheta = \beta_1\ddot{\delta} + \beta_0\dot{\delta} \tag{20}$$

where ϑ is the robot tilt angle, and δ is the wheel displacement angle with respect the robot body. The control input is assumed to be the wheel acceleration $\ddot{\delta}$. The model can be obtained from the general nonlinear model of the robot longitudinal dynamics, which has been derived by Antonello and Schenato (2017). For the design, the model is obviously reduced to the state–space form (1), after discretizing it with the exact method. The discretization is carried out with the sampling period T_s of the stabilizing controller (either emulation or model based), and it is equal to $T_s = 1\,\text{ms}$.

4. NETWORK CHARACTERIZATION

Shared wireless network dynamics is a random process and its theoretical modeling sets considerable challenges, see e.g. the work by Park et al. (2012) specifically devoted to IEEE 802.15.4. The characterization of the arrival processes, which are fundamental for control purposes, is not trivial since they depend on many factors as the channel quality and the channel status (idle/busy). Typical models like i.i.d. random variables, Gilbert-Elliot model, Markov chains, are not able to capture the behaviour of a Wi-Fi connection. For this reason, in this work, an experimental characterization is chosen to provide a quantitative description of the link quality, in terms of delay and packet loss statistics. Measurements have been conducted through the simplified hardware–in–the–loop experiment depicted in Fig. 3. Communication performance is assessed by comparing the sent and received packet sequences. Two machines are involved: the host PC (A – Linux x64) and a target board (B – Raspberry Pi 3 mod. B+), connected via Wi-Fi. Local clocks on both machines are synchronized with each other through the Network Time Protocol (NTP). Time–stamped UDP packets are exchanged between machine A and B, both in forward (A $\to$ B) and backward (B $\to$ A) direction. Sample frequency is set at 1 kHz (communication performances of Raspberry Pi 3 as a function of sampling frequency is available in Branz et al. (2019a)). The UDP receiver policy on machine B provides only the most recently received packet by repeatedly reading the incoming buffer and discarding older data. Differently, machine A saves up to 40 received packets in a custom buffer, starting from the most recently received one. The time–stamps in forward and backward branches are independent. The forward packet IDs are relayed back to A attached to the backward packets. Forward and backward delays are independently obtained by computing the time difference between transmission and reception time instants. The loop–back delay (A $\to$ B $\to$ A) is computed by tracking the packet IDs originally sent from A and relayed back from B. Link quality statistics are computed for forward, backward and loop–back paths. Forward and backward packet sequences are exploited in Sec. 5 to perform comparable numerical simulations of the system in different configurations, but under identical network conditions. The loop–back communication statistics are presented in this section to provide a quantitative assessment of the communication reliability.

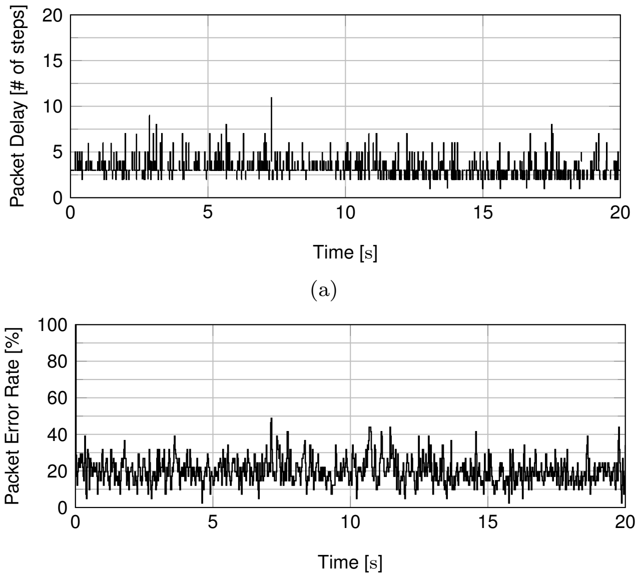

Fig. 4. Loop-back communication performances (example): (a) packet delay and (b) packet error rate (computed over 40 sample times).

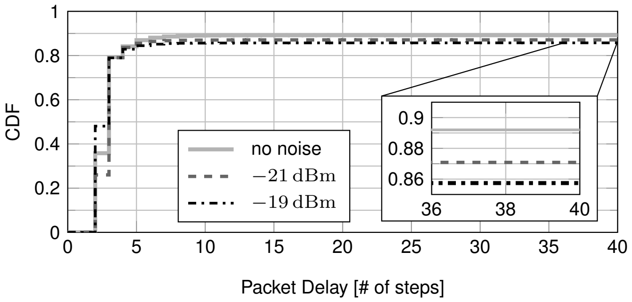

Fig. 5. Loop–back cumulative delay distribution at different noise levels (curves are averaged over 10 runs).

The host PC acts as access point of an IEEE 802.11n network on channel 6 with HT MCS bitrates enabled (index 7, 65 Mb/s, modulation 64-QAM 5/6) and the number of transmission retries set to 1. Experiments are conducted with a distance of approximately 3 m between A and B. In order to investigate the compatibility of the proposed approach with a real industrial environment, the connection is disturbed with a white gaussian noise from an Agilent E4432B signal generator with variable amplitude levels and 15 MHz of bandwidth around the 2437 MHz carrier. The noise is pulse–modulated with a period of 350 µs and a pulse width of 150 µs as suggested by Tramarin et al. (2016). The disturbance is emitted by a directional antenna pointing at the target board.

Table 1. Loop–back communication statistics (average over 10 runs, no noise)

	median	std	max	min
packet delay [# steps]	3	0.96	20	2
PER [%]	10.82	2.43	42.38	2.34

In addition, the laboratory environment is crowded with other networks, possibly operating on the same channel.

An example of delay and Packet Error Rate (PER) sequences is shown in Fig. 4 for the loop–back case. Figure 5 presents the Cumulative Distribution Function (CDF) of loop–back packet delay. The three curves are averaged over 10 runs and refer to three different noise levels (no noise, -21 dBm and -19 dBm). Increasing injected noise causes a small degradation of communication performances and, as a general note, average packet loss is always above 10 %. Table 1 summarizes the loop–back communication performances in terms of packet delay and PER.

5. SIMULATIONS

The dynamics of the controlled system described in Sec. 3 is simulated to validate the system performances and to compare the proposed Kalman estimator with a solution based on a simple observer (namely, a reduced-order observer where the wheel and body angular rates $\dot{\delta}$ and $\dot{\vartheta}$ are obtained by simple differentiation of the corresponding angular measurements). Simulations are performed entirely on the host PC. The complete simulation model is depicted in Fig. 6a. The nonlinear plant dynamics is simulated including the low–level speed control loop, the feed–forward compensation of friction and the actual hardware models (i.e. motors driver and sensors). The state observer (either Kalman filter or simple observer) and the balance control perform state estimation and compute the acceleration reference for the speed control. The Wi-Fi connection between the two sides is simulated exploiting the measured delay sequences collected with the hardware–in–the–loop tests described in Sec. 4: the transmitted signal (either the command or the measure) is temporarily stored in a buffer and delivered after the corresponding delay available from experimental data. In this way, simulations guarantee a fair comparison between alternative configurations, allowing to study the performances of different approaches under identical and realistic network conditions.

Initial position conditions are $\vartheta_0 = 5\,\text{deg}$ and $\gamma_0 = 0\,\text{deg}$; initial velocity is zero. While holding the vertical position, the system is required to track a step reference to the wheel angle γ, determining a translation of the vehicle position of 0.1 m. The step occurs at $t = 15\,\text{s}$. Simulation results are shown in Fig. 7. The system response is presented comparing the emulation–based and the model–based approaches. The example case depicted in the plots on the right shows that that proposed solution allows to control the system in a wider range of external conditions, proving higher tolerance on packet delays and losses. The proposed Kalman filter provides the controller with a more reliable state estimate, thus allowing to keep the vertical equilibrium when the simpler reference configuration catastrophically loses control. The presented results show that the adoption of the model–based estimator allows to guarantee stability when the simple observer fails to do so due to the presence of a disturbed communication link.

6. CONCLUSION

The problem of control over wireless is tackled in this work by adopting the IEEE 802.11 (Wi-Fi) protocol as an alternative to conventional wireless protocols for automation. Wi-Fi theoretically allows to achieve considerably higher data rates, at the cost of reduced reliability (i.e. packet delays and losses). Management of the intrinsic unreliability of Wi-Fi is of key importance to make it a suitable candidate for the implementation of wireless control. This work suggests the combined use of Wi-Fi and advanced state estimation techniques to achieve wireless control at 1 kHz. Additionally, common embedded systems have been adopted, aiming at the validation of the proposed solution on simple commercial hardware.

Focus is on the validation of a time–variant Kalman filter implementation that takes advantage of a buffer of received data to deal with communication flaws. To minimize customization of commercial hardware, communication protocols are adopted with no modifications, except for the development of a time–efficient packet receiving policy. A Segway–like vehicle is adopted as benchmark application. The proposed solution is compared to a simple observer in relevant hardware–in–the–loop experiments, under variable external noise condition. Experiment shows that the proposed approach allows to keep the vertical balance, while the simpler reference configuration fails due to excessive packet delays. Stability is achieved in a wider range of external conditions, highlighting higher tolerance on packet delays and losses. Results prove the advantages of exploiting the proposed solution, showing relevant performance improvements on actual hardware and paving the way to industrial applications. Future developments of the current research include the experimental validation of the proposed method on the real balancing robot setup.

REFERENCES

Antonello, R. and Schenato, L. (2017). Longitudinal state–space control of the balancing robot. Technical report, University of Padova, Dept. of Information Engineering. URL `http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/SEGWAY_GUIDE.pdf`.

Branz, F., Antonello, R., Tramarin, F., Fedullo, T., Vitturi, S., and Schenato, L. (2019a). Embedded systems for time–critical applications over Wi-Fi: design and experimental assessment. In *Proceedings of IEEE International Conference on Industrial Informatics (INDIN'19)*.

Branz, F., Pezzutto, M., Antonello, R., Tramarin, F., and Schenato, L. (2019b). Drive-by-Wi-Fi: testing 1 khz control experiments over wireless. In *2019 18th European Control Conference (ECC)*, 2990–2995. IEEE.

Garone, E., Sinopoli, B., and Casavola, A. (2008). LQG control over lossy TCP-like networks with probabilistic packet acknowledgements. In *2008 47th IEEE Conference on Decision and Control*, 2686–2691. IEEE.

Imer, O.C., Yüksel, S., and Başar, T. (2006). Optimal control of lti systems over unreliable communication links. *Automatica*, 42(9), 1429–1439.

Lin, H., Su, H., Shi, P., Shu, Z., Lu, R., and Wu, Z.G. (2017). Optimal estimation and control for lossy network: stability, convergence, and performance. *IEEE Transactions on Automatic Control*, 62(9), 4564–4579.

Lin, H., Su, H., Shu, Z., Wu, Z.G., and Xu, Y. (2015). Optimal estimation in udp-like networked control systems with intermittent inputs: Stability analysis and suboptimal filter design. *IEEE Transactions on Automatic Control*, 61(7), 1794–1809.

Maass, A.I., Nešić, D., Postoyan, R., Dower, P.M., and Varma, V.S. (2017). Emulation-based stabilisation of networked control systems over WirelessHART. In *2017*

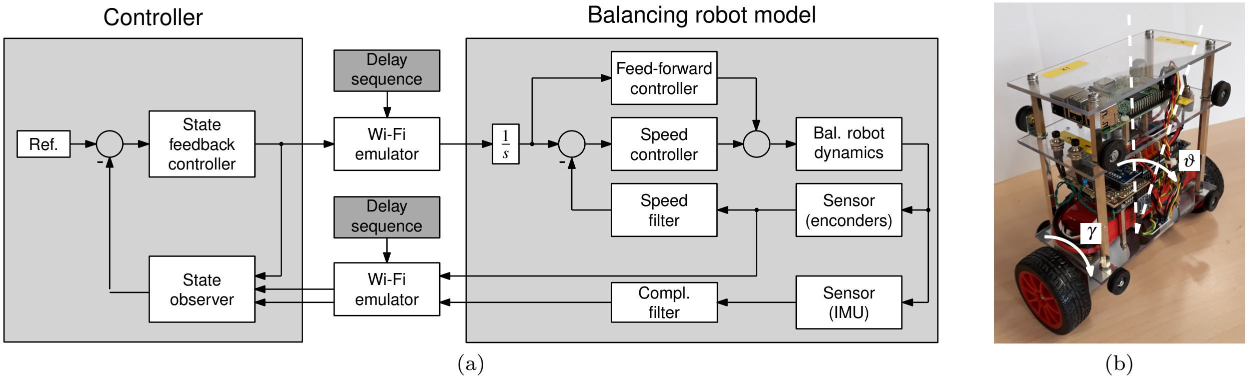

Fig. 6. Simulation model architecture: (a) overall block diagram; (b) balancing robot reference prototype (upon which the numerical simulations are based).

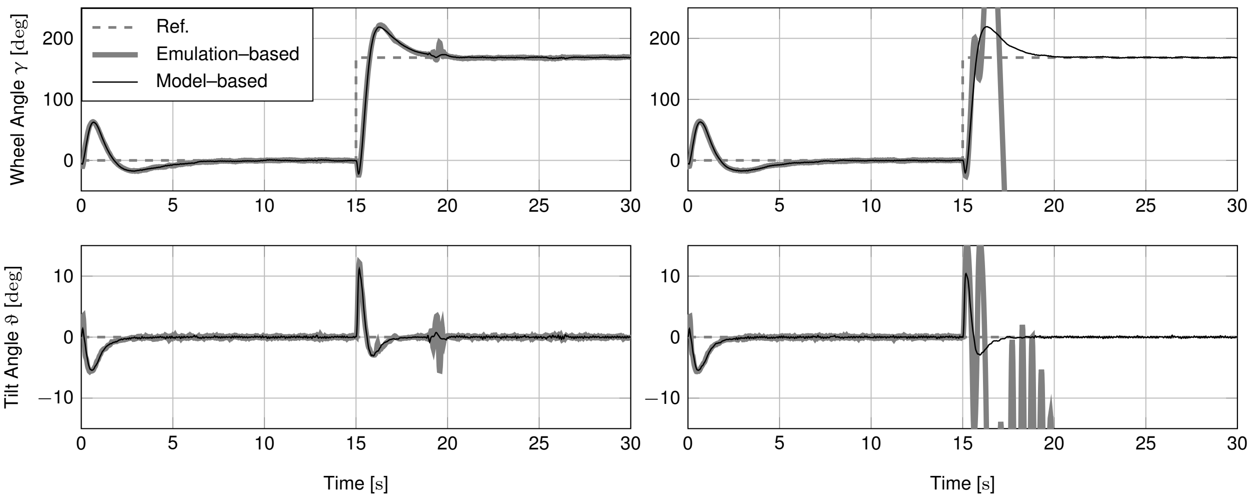

Fig. 7. Simulation results: tilt angle (top) and wheel angle (bottom) in two different simulation runs (left and right).

IEEE 56th Annual Conference on Decision and Control (CDC), 6628–6633. IEEE.

Nešić, D. and Teel, A.R. (2004). Input-output stability properties of networked control systems. *IEEE Transactions on Automatic Control*, 49(10), 1650–1667.

Park, P., Di Marco, P., Fischione, C., and Johansson, K.H. (2012). Modeling and optimization of the IEEE 802.15.4 protocol for reliable and timely communications. *IEEE Transactions on Parallel and Distributed Systems*, 24(3), 550–564.

Petersen, S. and Carlsen, S. (2011). WirelessHART versus ISA100.11a: The format war hits the factory floor. *IEEE Industrial Electronics Magazine*, 5(4), 23–34.

Schenato, L. (2008). Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Transactions on Automatic Control*, 53(5), 1311–1317.

Schenato, L. (2009). To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54(5), 1093–1099.

Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S.S. (2007). Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1), 163–187.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., and Sastry, S. (2008). Optimal linear lqg control over lossy networks without packet acknowledgment. *Asian Journal of Control*, 10(1), 3–13.

Tramarin, F., Vitturi, S., Luvisotto, M., and Zanella, A. (2016). On the use of IEEE 802.11 n for industrial communications. *IEEE Transactions on Industrial Informatics*, 12(5), 1877–1886.

Walsh, G.C., Ye, H., and Bushnell, L.G. (2002). Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10(3), 438–446.

Xia, Y.Q., Gao, Y.L., Yan, L.P., and Fu, M.Y. (2015). Recent progress in networked control systems—a survey. *International Journal of Automation and Computing*, 12(4), 343–367.

Zhang, X.M., Han, Q.L., and Yu, X. (2016). Survey on recent advances in networked control systems. *IEEE Transactions on Industrial Informatics*, 12(5), 1740–1752.

Zhu, X., Lin, T., Han, S., Mok, A., Chen, D., Nixon, M., and Rotvold, E. (2012). Measuring WirelessHART against wired fieldbus for control. In *IEEE 10th International Conference on Industrial Informatics*, 270–275.