An Integrated System Simulation Approach for Wireless Networked Control Systems

Peter Horvath, Mark Yampolskiy, Yuan Xue, Xenofon D. Koutsoukos and Janos Sztipanovits
Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN, USA

***Abstract*—Cyber-Physical Systems (CPS), such as networked control systems, are increasingly deployed over wireless networks. Given the sensitivity of control systems to networking conditions such as packet drops, delays and jitters, it is important to verify and evaluate the control system properties under realistic wireless networking deployment scenarios. However, current research is often based on simplistic models of the wireless network physical layer behaviors. In this paper, we point out deficiencies in the existing simulation methods for the performance evaluation of wireless networked control systems and present a novel simulation framework for wireless network control systems. Our approach aims at capturing the effects in the physical layer more accurately than state-of-art simulators are capable of. An integrated simulation tool, based on open-source solutions, is presented and a case study of a networked control system is also provided to illustrate the capabilities of our simulation tool.**

I. Introduction

Cyber-physical systems often incorporate control loops that are closed over wireless networks. Accurate modeling of these communication networks is essential in the design and verification of CPS. Controller design needs to take into account the wireless channel characteristics (e.g., delays, packet losses). Completed control designs need to be evaluated under realistic network conditions before deployment to prove that they are resilient under the impairments they would possibly encounter. Performance evaluation of wireless networks, either analytically or by simulations, poses a complex problem in itself. Integrating wireless network modeling and simulation of networked control systems is even more challenging.

Looking at short-range wireless networking protocols, mostly IEEE 802.11 and 802.15.4 are considered in the CPS context. Performance evaluation of these system is possible either using analytic models or by resorting to simulations. An impressive effort has been made over the last 15 years in the area of analytical performance modeling for these technologies. For example, Markov chain based approaches have been widely adopted by networking experts. Inevitably, often simplified assumptions need to be made in this approach in order to maintain tractability of the models. Such common assumptions refer to I) statistics of packet losses, and II) the statistics of the traffic pattern.

Regarding the packet loss, assumptions range from the completely error-free channel to more realistic error models taking some kind of transmission range of the nodes into account. Some approaches address the drawbacks of the fixed transmission/interference range by considering a "transitional region" between "connected" and "disconnected" states [1]. Some also account, to some extent, for the link budget and the dependence of the packet error rate on the instantaneous signal-to-noise ratio.

On the other hand, theoretical models frequently make assumptions on the traffic statistics. Often Poisson traffic is assumed which is clearly not realistic between a single pair of nodes forming a sampled networked control system in which fixed-size packets need to be transmitted in more-or-less regular intervals. Thus, despite the existence of closed form methods for the problem under investigation with simplifying assumptions, it is still desirable to verify the robustness of the proposed schemes and algorithms under more realistic assumptions, resorting to simulations. We also emphasize that as wireless systems become more mature and complex, adopting advanced physical layer technologies, like multiple input, multiple output (MIMO) antenna processing, advanced interference cancellation schemes, good simulation models become indispensable to be able to assert the effects of various system parameters.

When it comes to to simulation approach, widely available packet-level simulators are capable of simulating the most important wireless network technologies, although with implementation-dependent accuracy when it comes to modeling the layer 1 and 2 effects. However, these tools generally operate employing a very detailed message exchange model spanning over multiple ISO/OSI layers. This detail of implementation makes them less than ideal for applications where *scalability* is required.

In this paper, we borrow the idea of *system-level simulation* from the wireless cellular community and propose a similar simulation approach in the context of wireless networked control systems. The system simulation approach strives to model the whole system at a higher level of abstraction than link-centric models, but encompassing the entire network as the highest level of hierarchy. The higher abstraction level generally yields to gains in the run-time requirements, provides improved scalability and the system-level view enables fairly accurate modeling of inter- and intra-system interference. If the link abstractions are constructed appropriately, then the accuracy of the system-level model is also satisfactory and enables taking the performance of modulation and coding schemes, and effects of various link impairments into account.

We also outline a novel open-source simulation tool that adopts the principle of system-level modeling for wireless networked control systems in an efficient, general and scalable way. We discuss the implementation and validation methods for the tool. Finally, the capabilities of our proposed tool will be illustrated using a simple application example.

The organization of this paper is as follows. Section III proposes a methodology for capturing the interactions between the lower layers and the control applications in a potentially complex wireless networked control system. Section IV describes the simulation tool being developed that follows the outlined methods. Finally, section V presents the case study using our tool for a simple networked control system.

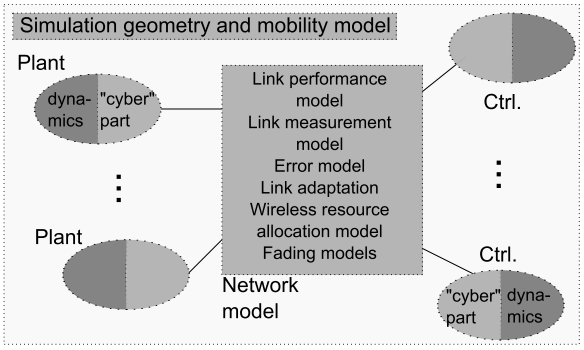

Fig. 1. System simulator for networked control systems

II. Related work

Many network simulators, including the popular *ns-2* or OMNeT++ and the various frameworks of the latter, contain very detailed packet-level simulation models of the most widely used internetworking protocols. They also aim at modeling relevant wireless technologies, notably the IEEE 802.11 family and IEEE 802.15.4. The amount of details varies depending on the simulators or their frameworks. But, generally, important effects in the physical layer (fading of the signal, shadowing effects, realistic models of path loss, eventual channel codes, advanced retransmission schemes) are usually neglected [2]. These tools inherently adopt a link-centric view of the network, and the system-level implications (prominently the interference caused to/suffered from other nodes and/or other networks) are often modeled in less realistic ways.

According to the literature, directly extending network simulators with control studies has not found widespread application, which is likely to be attributed to the lack of scalability of these simulators. More widespread is the hybrid approach, in which the network simulator and the simulator for the system dynamics are handled by specialized tools, and the inter-domain synchronization is handled by a separate entity. As the cyber part typically uses discrete event semantics while the physical part obeys continuous time or discrete time semantics, appropriate synchronization is necessary between these domains of computation. To mention a few such representative examples, [3] integrates Simulink for the physical part and *ns-2* for the cyber part, and relies on an innovative High-level Architecture (HLA)-based scheme to coordinate between the constituent simulators. Similarly, the co-simulation platform [4] employs *ns-2* for network simulation, whereas Modelica is used to simulate the physical parts. Finally, the Matlab-based TrueTime engine [5] supports a fairly sophisticated simulation of wireless networked control systems. However, TrueTime makes some simplifying assumptions to the packet loss model, fading model. Most notably, TrueTime doesn't model the link adaptation scheme in 802.11, i.e. it assumes a fixed data rate irrespective of the actual channel conditions, which gives pessimistic results for example in 802.11g where the lowest data rate is 1 Mbit/s, and the highest is 54 Mbit/s. Current TrueTime versions support only the contention-based access mechanism for 802.15.4.

III. The system simulation approach

The system-level simulation approach has been widely used to assess the impacts of link-level behavior on the end-to-end system-level performance of networks that might consist of multiple base stations and possibly hundreds or thousands of mobile stations (see [6], [7] for applications to state-of-art mobile standards).

A basic block diagram of the proposed simulation approach is shown in Fig. 1. Plants and controllers constitute of the continuous-time or discrete-time dynamics, and a simplified network interface that models the application-layer processing of the packets.

The treatment of the physical and link layer within the simulator differs from the traditional approach. In the traditional packet-oriented simulations, stations are treated as separate entities, typically using composition to model the radio protocols, MAC functionality etc., trying to accurately model the message exchange between the nodes. Thus, this approach yields to a link-oriented modeling approach.

Our simulator, on the other hand, focuses on the performance aspects, abstracting the exact internal procedures of the lower layers, and not strictly following the traditional compositional approach when modeling the functionality of the network. It is, however, important that the abstraction doesn't compromise the accuracy of the model. One single channel entity oversees the communication between all network nodes. This allows a system-oriented treatment in which it is possible to introduce spatially correlated shadowing, correlated probabilistic fading, explicit modeling of layout- and traffic-dependent interference etc.

Application traffic models and higher-layer protocol layers are also taken into account using explicit models. Regarding the physical layer, the data about the geometrical arrangement of the stations is maintained (e.g., for calculating the distance between stations, determining the gains of directional antennas, setting the Doppler spread of the channel fading processes depending on the relative velocity and direction of movement of stations etc.). Transmitters are characterized by their powers, receivers are commonly characterized by

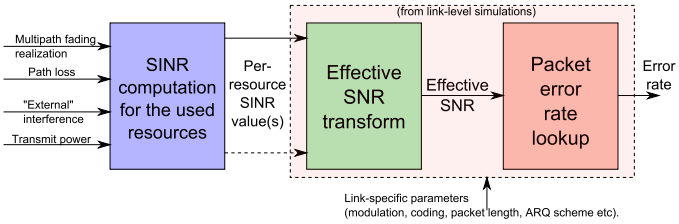

Fig. 2. Determination of packet error probabilities

their noise figure. The effects of the wireless channels are taken into account using realistic channel fading models at three different levels: I) a distance-dependent average signal attenuation ("distance loss") influences the average attenuation between transmitter and receiver, II) *shadowing* effects might be considered, which reflect random, but temporally and spatially correlated fluctuations around the local mean attenuation, and III) the time-dependent multipath ("fast") fading that is due to random superposition of multipath rays and changes on a much faster time/distance scale than the first two factors.

In general, the fading processes are generated according to some well-known channel model that describes well the actual propagation scenario, and, preferably, realizations of which are computationally inexpensive to generate. The link measurement model determines the measurements that will eventually be used for link adaptation, and the link performance model, which predicts the probability of a transmission error under the current link conditions.

The system simulator determines, based on the link error model, the packet error probability for the transmitted packets. This is schematically illustrated in Fig. 2. Instead of performing extremely time-consuming Monte Carlo simulation for every transmitted packet, existing system simulators tend to abstract the physical layer and predict the error probability based on some computationally inexpensive mapping from the current signal-to-noise (plus interference) ratio to the packet error ratio using look-up tables that have been generated using one-time detailed link-level simulations. Some possible mappings will be mentioned later in conjunction with our implementation. Mappings are selected such that they can also handle the case when the data packet experinces a multi-state channel, i.e., the channel changes significantly during data transmission, or the data is transmitted using multicarrier modulation over a frequency-selective wireless channel and different subcarriers experience different amounts of attenuation due to multipath phenomena.

The proposed simulation methodology is most easily adaptable to systems which are designed to keep the stations using orthogonal resources (for example, in a scheduled time-division manner) instead of using random access for every transmission attempt. A prominent example of such a system is WirelessHART, in which intra-system interference and random transmission delay is minimized by using a centralized time-division frame structure.

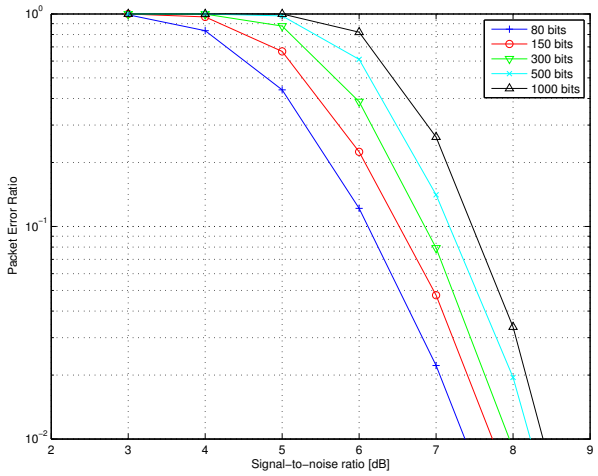

Fig. 3. IEEE 802.15.4 packet error ratios for different payload lengths

IV. The CPS simulation tool

A. Simulation principles

A simulation tool has been developed according to the principles outlined in the previous section. Currently, the IEEE 802.15.4 [8] model is implemented in detail. Among others, both ZigBee and, to some extent, WirelessHART devices rely on the physical layer and data link layer specified in this standard [9], which makes it especially relevant for the CPS community. Contention-based (Carrier-Sense Multiple Access with Collision Avoidance) as well as contention-free (Guaranteed Time Slot, GTS) access mechanisms are implemented, as descibed in the standard. The present network layer is rather simple, supporting star-like topologies, with its behaviour closely resembling that of the network layer of ZigBee.

In order to determine the packet loss probability for a given packet, one needs the mapping from the effective SNR to the packet error probability. As an example, Fig. 3 shows the packet error ratio versus signal-to-noise ratio over additive white gaussian noise (AWGN) channel for the IEEE 802.15.4 2.4 GHz PHY waveform for various payload lengths, including the MAC overhead. These curves have been obtained using our proprietary Monte-Carlo simulations using an appropriate link-level simulator, and are stored in look-up tables in the CPS simulator itself.

In the simplest setting, these curves can be directly used to determined the packet error ratio if the fading in the channel can be regarded approximately constant during the transmission of one block (quasi-static assumption), and the channel magnitude response is approximately constant over the transmission bandwidth (frequency-flat fading). If the latter assumption is not valid, the effects of the frequency selectivity need to be taken into account, which also depends on the signal processing algorithms employed in the respective receiver to cancel, or even exploit the distinct multipath echos in the channel. This presents an additional non-trivial step in the performance evaluation for IEEE 802.15.4-like, relatively wideband, spread spectrum-based air interfaces. In cellular

context, this challenge is usually tackled by applying various semi-empirical SNR penalties to the instantaneous SNR. The most efficient way of implementing this feature for such simulations is still under consideration.

At this point it should be noted that finding the packet loss probability, given an instantaneous channel realization and co-channel interference value, can be done in a more general way if the communication scheme employs coded orthogonal frequency division multiplexing (OFDM), as for example certain transmission modes of 802.11 do. OFDM, which uses low-rate parallel transmission on multiple subcarriers, "converts" the frequency-selective wireless channel into a set of parallel frequency-flat fading channels, it is relatively simple to find an compression function that yields a scalar effective signal-to-noise ratio based on the per-subcarrier signal-to-noise-and-interference ratios (SINRs). If the compression function is chosen appropriately, the resulting effective signal-to-noise ratio marks a point on the AWGN packet error ratio curve (as in Fig. 3) which approximates well the real packet error ratio that the multipath fading channel were exhibiting. Such mappings include exponential effective SINR mapping and mean instantaneous capacity mapping [7].

B. Simulator implementation

The present simulator, unlike the ones presented in [3] and [4], is monolithic and completely implemented in the Python programming language. Due to its monolithic nature, the overall simulation time tends to be significantly lower compared to the hybrid simulators. The simulations are governed by the SimPy discrete-event, process-based simulation engine [10]. The process-oriented discrete event framework differs from the more conventional event-oriented simulation approach primarily adopted in specialized network simulators (*ns-2, OMNeT++*). Although the event-oriented approach is regarded superior within the networking community, and SimPy itself doesn't ship with any specialized networking code, it has actually been proved an extremely powerful means for our purposes. The process-based nature of SimPy, however, enables straightforward simulation of event-driven control systems, which is a significant advantage over some existing tools. SimPy provides monitors for observing simulation events, and interfaces well with NumPy/SciPy for numerical evaluation and presentation of simulation results.

The physical part of the system is currently also simulated in Python. Both plants and controllers can be described by continuous-time state-state equations, or the controller can be treated as a discrete-time system, similarly to the assumptions made in TrueTime. The continuous-time state-space equations are numerically solved using the Assimulo Python package [11], which, in turn, is a wrapper around the C-based SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers) package [12]. It is foreseen that the possibility of using Simulink-generated plant models will be added.

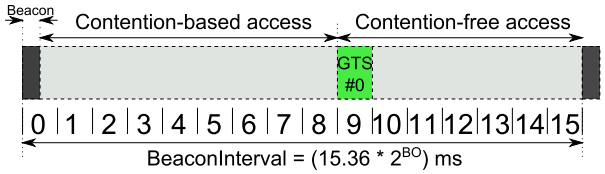

Fig. 4. Time slot structure in 802.15.4

C. Tool validation

We validate the two main components of the tool, the modeling of control dynamics and the network model, separately. Then the overall functionality is verified similarly to the approach presented in [4]: a single plant-control pair is connected over non-lossy two-way channel with fixed delay. The plant is a first-order dynamic system, the controller is a simple proportional one, and the plant output is compared to the known analytic solution.

In addition, the performance of the IEEE 802.15.4 2.4 GHz PHY/MAC model is compared to simulation results from the literature assuming loss-free operation with Poisson packet arrival statistics for example presented in [13]. In addition, we present a few simulation results purely for the network part. We focus on the contention-free GTS mechanism which is often regarded as a better means for delay-sensitive applications than the more widespread contention-based (CSMA/CA) access scheme within 802.15.4. We assume beacon-enabled mode, in which the coordinator transmits a regular beacon at a given interval, marking the limits of a superframe consisting of 16 slots. One of the timeslots is the GTS assigned to the simulated packet source, while the packet destination receives the 1-byte packets transmitted by the source.

As shown in Fig. 4, the superframe structure can be described by the protocol constant *Beacon Order (BO)*. Increasing BO results in exponential increase of the superframe duration, and at the same time the slot capacity also increases exponentially. We assume 100 % duty cycle, i.e., there is no inactive period within the superframe, which can be configured setting the protocol constant *Superframe Order (SO)* equal to BO.

Acknowledged mode is in use with at most three layer-2 retransmissions, after which the packet is dropped. The queue size on the transmit side is one packet, which is realistic in the NCS setting. We consider three different channel models: lossless channel as baseline, Bernoulli packet loss channel, over which the packet loss is independent with a loss probability of 0.2, and temporally correlated frequency-flat Rayleigh fading [14], assuming a moving terminal with a velocity of 1 m/s, respectively. In the Rayleigh fading case the average SNR is adjusted such that the long-term average packet loss probability is approximately 0.2 as in the Bernoulli case, but the loss probabilities are not independent due to the temporal fading correlation.

To illustrate the effects of the channel model on the distribution of the packet delays, which is crucial when networked control is considered, Fig. 5 shows the cumulative distribution

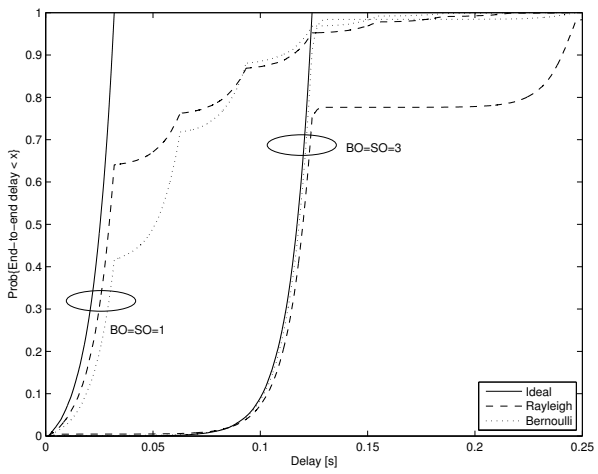

Fig. 5. CDF of the end-to-end packet delay in IEEE 802.15.4 GTS, with different channel models (Poisson arrivals, mean IAT=0.01 s)

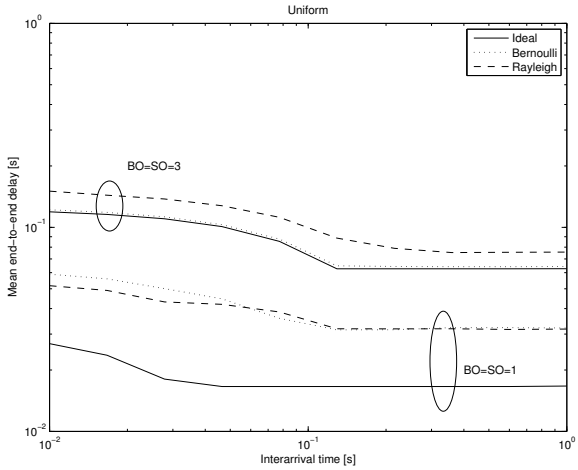

Fig. 6. Mean end-to-end packet delay in IEEE 802.15.4 GTS, with different channel models (Uniform arrivals)

function of the end-to-end delay, assuming only one slot is allocated in the GTS between the packet source and destination nodes, assuming Poisson traffic for two different BO values. It is interesting to contrast the Bernoulli and Rayleigh model for the BO=3 case. For the Bernoulli case, the delay is essentially bounded in one superframe period, whereas for the Rayleigh channel, a significant number of packets needs two superframes to get through. This is attributed to the lack of any interleaving and error control coding within the 802.15.4 PHY: if fading only changes slowly over time, stations stuck in fading minima experience prolonged outage intervals.

Similarly, the *mean* end-to-end delays are shown in Fig. 6, now assuming uniform interarrival times for two different BO settings. It can be seen that the parameter choices in the link layer and the underlying physical channel give rise to quite different transmission behavior. Hence the mean delay strongly depends on the traffic load, protocol parameters and channel model.

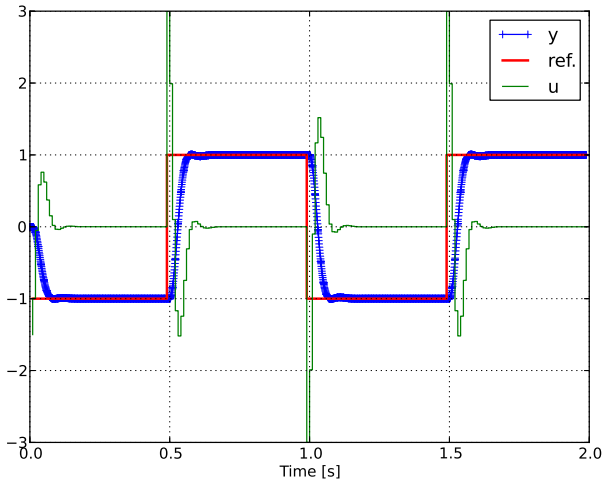

Fig. 7. Output of the linear system, ideal transmission. y is the measured output, u is the control input, and ref denotes the reference signal.

V. Case Study

The present application example features an event-driven discrete-time controller instead of the continuous-time one mentioned in the validation results. The setup is similar to one of the TrueTime demonstration examples: an unstable continuous plant with transfer function

$$H(s) = \frac{1000}{s^2 + s}$$

is controlled using a discrete PD regulator over a wireless network. The controller is event-driven. The sample time of the sensor/actuator node is 10 ms, and the payloads are assumed to be 40 bits in size. For reference, the plant output signal y and control signal u are shown in Fig. 7 for the case when the plant and controller are directly connected, i.e., there are no delays or losses in the control sytem.

In Fig. 8 the same system is controlled over an IEEE 802.15.4 link in acknowledged mode. To cope better with network-induced delays, a sligthly modified discrete PD controller (obtained by setting $K = 0.8, T_d = 0.04$ s) is employed. Both the plant output and the control signal get transmitter over the wireless link in one of the three guaranteed time slots assigned for each node. The GTSs are spread as evenly as possible over the entire 802.15.4 superframe duration, at the same time observing the rules described in the standard for GTS allocation. For the superframe duration, BO=SO=0 have been chosen (lowest slot capacity, highest slot recurrence rate) in order to minimize latency over the wireless network. As evident from the figure, protocol delays clearly affect the control performance.

As shown in Fig. 9, after introducing the channel impairments, the performance degrades seriously. In this case, the channel introduces pure, temporally correlated Rayleigh fading according to the standard Jakes' model, where we assume that the Doppler spread is 4 Hz (this can be physically the case if one of the terminals is moving at approx. 1 km/h, or objects in the environment move). The channel coherence

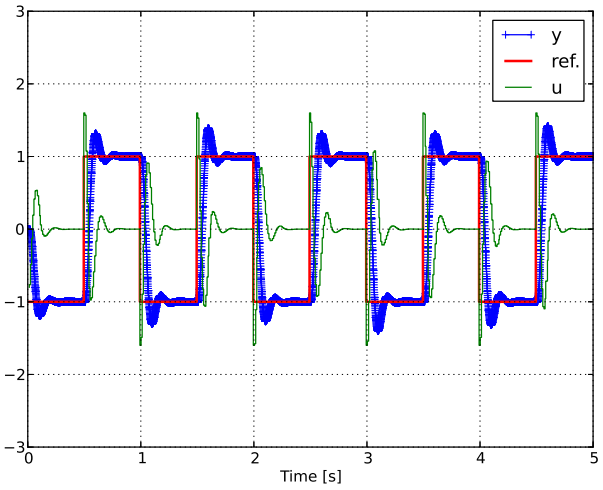

Fig. 8. Output of the linear system, IEEE 802.15.4 channel without impairments.

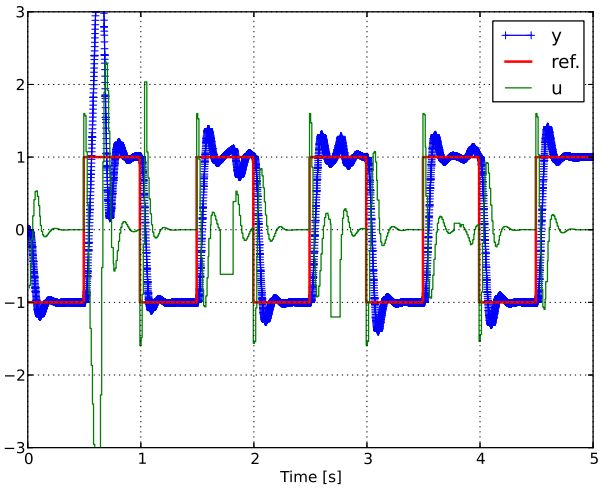

Fig. 9. Output of the linear system, IEEE 802.15.4 over correlated Rayleigh fading channel

time is around 100 ms, i.e. the channel state shows significant correlation over this time. Fading gives rise to a long-term packet loss rate of approx. 7 % in the 802.15.4 physical layer. As 802.15.4 doesn't possess any error control capability beyond the layer-2 acknowledgement/retransmission, channel-introduced errors are likely to show a similar correlated behaviour, i.e. consecutive error events are not independent. This can indeed be observed in Fig. 9.

The simulation time required for this simulation is less than the real time using an off-the shelf workstation PC, although no efforts have yet been made to optimize the underlying networking code. This is in contrast with the ns-2 based tools that require orders of magnitudes more to solve the same problem.

VI. Conclusion

In this paper we pointed out a potential in modeling wireless networked control systems more accurately by using the system simulation approach. The most important physical layer impairments have been summarized, and a modeling methodology was outlined that is based on the link-to-system mapping approach. The proposed framework is implemented in our standalone, open-source CPS simulation tool. The presented simulation methodology is best suited for accurate study of impairments of the wireless propagation in systems with predominantly orthogonal resource usage.

As future work, we intend to use the tool for cross-layer system design purposes. The modeling approach lends itself to study the effects of various modulation and coding schemes, retransmission strategies, link adaptation algorithms etc. on the control performance.

Acknowledgment

This work is supported in part by the National Science Foundation (CNS-1035655, CCF-0820088), U.S. Army Research Office (ARO W911NF-10-1-0005) and Lockheed Martin. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

[1] M.-H. Zayani, V. Gauthier, and D. Zeghlache, "A joint model for IEEE 802.15.4 physical and medium access control layers," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, 2011, pp. 814–819.

[2] M. Dohler, R. W. Heath, A. Lozano, C. B. Papadias, and R. A. Valenzuela, "Is the PHY layer dead?" *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 159–165, 2011.

[3] D. Riley, E. Eyisi, J. Bai, X. Koutsoukos, Y. Xue, and J. Sztipanovits, "Networked control system wind tunnel (NCSWT) – an evaluation tool for networked multi-agent systems," in *4th International ICST Conference on Simulation Tools and Techniques (SIMUTools 2011)*, Barcelona, Spain, March 2011.

[4] A. T. Al-Hammouri, "A comprehensive co-simulation platform for cyber-physical systems," *Computer Communications*, January 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412000047

[5] J. Eker and A. Cervin, "A Matlab toolbox for real-time and control systems co-design," in *Proceedings of the 6th International Conference on Real-Time Computing Systems and Applications*, 1999, pp. 320–327.

[6] J. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of LTE networks," in *IEEE 71st Vehicular Technology Conference (VTC 2010-Spring)*, May 2010, pp. 1 –5.

[7] "WiMAX system evaluation methodology," WiMAX Forum, Tech. Rep., 2008.

[8] *IEEE Std 802.15.4-2006, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Computer Society Std.

[9] F. Xia, R. Hao, Y. Cao, and L. Xue, "A survey of adaptive and real-time protocols based on IEEE 802.15.4," *International Journal of Distributed Sensor Networks*, p. 11, 2011.

[10] *SimPy Simulation Package*. [Online]. Available: http://simpy.sourceforge.net/

[11] "Assimulo." [Online]. Available: http://www.jmodelica.org/assimulo

[12] "SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers)." [Online]. Available: https://computation.llnl.gov/casc/sundials/main.html

[13] F. Chen, N. Wang, R. German, and F. Dressler, "Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications," *Wireless Communications and Mobile Computing*, vol. 10, no. 5, pp. 609–621, 2010. [Online]. Available: http://dx.doi.org/10.1002/wcm.736

[14] B. Azimi-Sadjadi, D. Sexton, P. Liu, and M. Mahony, "Interference effect on IEEE 802.15.4 performance," in *International Conference on Networked Sensing Systems*, Chicago, IL, 2006.