

NCSbench Demo: Reproducible Benchmarking Platform for Networked Control Systems

Samuele Zoppi*, Onur Ayan*, Fabio Molinari[†], Zenit Music[†], Sebastian Gallenmüller[‡], Georg Carle[‡], and Wolfgang Kellerer*

*Chair of Communication Networks, Department of Electrical and Computer Engineering, TUM, Germany

[†]Control Systems Group, Department of Electrical Engineering and Computer Science, TU Berlin, Germany

[‡]Chair of Network Architectures and Services, Department of Informatics, TUM, Germany

Email: {samuele.zoppi, onur.ayan, sebastian.gallenmueller, carle, wolfgang.kellerer}@tum.de, molinari@tu-berlin.de, zenitmusic@mailbox.tu-berlin.de

Abstract—Cyber-Physical Systems (CPS) are widely spreading thanks to fast-paced technological breakthroughs of micro-controllers and communication networks. Among them, Networked Control Systems (NCS) gained notable attention thanks to their application in industrial operations. In NCS, the interconnection of a control logic with sensor and actuators used to steer a physical system occurs over a communication network. Despite large research interest on NCS, the reproducibility and comparison of experimental results are difficult to achieve. This is caused by the lack of well-established models and methodologies that combine the theoretical and practical aspects of NCS. We tackle this problem by proposing and demonstrating NCSbench: the first open-source reproducible benchmarking platform for NCS. NCSbench enables the benchmarking of research experiments using a networked two-wheeled inverted pendulum robot. For each benchmarking experiment, a set of values is measured and used to quantify the key performance indicators (KPIs) of the NCS. In our demonstration, we visualize in real-time the evolution of the benchmarking KPIs on a web-based Graphical User Interface.

Index Terms—CPS, NCS, Open-source, Benchmarking, Reproducibility, KPI, QoS, Delay, Packet loss, QoC, Real-time, GUI.

I. INTRODUCTION

Cyber-Physical Systems (CPS) are widely spreading thanks to fast-paced technological breakthroughs of micro-controllers and communication networks [1]. A CPS consists of the interconnection of sensors, a control logic, and actuators together interacting over a physical system. When the logic belongs to an automatic control system, CPS are called Networked Control System (NCS) [2]. Over the last decade, NCS gained popularity thanks to their application in industrial operations, for instance, in the closed-loop regulatory control of production machines [3].

Despite the large number of results available in the literature [4], [5], reproducibility and comparison of NCS experimental results are difficult to achieve. This problem originates from the lack of well-established models and methodologies that combine the theoretical and practical aspects of NCS for different research communities: control systems, computation systems, and networking. Therefore, the literature shows an extensive variety of hardware and software platforms, experimental methods, and investigated performance metrics which

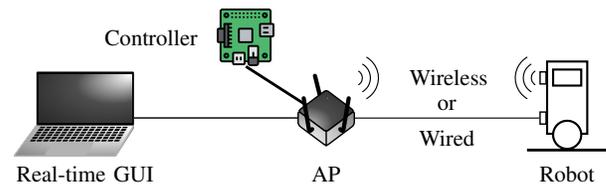


Fig. 1: Experimental setup of the demonstrated NCSbench platform. The demonstration supports two network interfaces: an Ethernet adapter or a 2.4 GHz W-LAN dongle.

are difficult to combine. Furthermore, existing studies mainly focus on specific aspects of the NCS and do not provide a complete characterization.

We tackle this problem by proposing and demonstrating NCSbench: the first open-source¹ reproducible benchmarking platform for NCS [6]. NCSbench enables the *benchmarking of research experiments* using a networked two-wheeled inverted pendulum robot, which has a long tradition in literature [7], [8] and industry [9]. The result of each benchmarking experiment is a set of *key performance indicators* (KPIs) used to monitor the operation of the NCS. In our demonstration, we visualize in real-time the evolution of the benchmarking KPIs on a web-based Graphical User Interface (GUI).

The NCSbench platform was developed keeping in mind reproducibility design principles to be replicated and deployed for arbitrary research purposes. This flexibility is achieved via open-source software and low-cost and highly accessible hardware components. NCSbench is built using the Lego Mindstorms™, communicates using standard Ethernet and W-LAN network interfaces, is open-source¹, and is programmed using Python. This way, all the elements of the NCS architecture of can be easily changed. Different physical plants can be built using Lego™, new control logics can be programmed in Python, arbitrary TCP/IP network interfaces can be connected, and the most popular computing systems and operating systems can be used.

In the next section, we provide a detailed description of the NCSbench demonstration and its experimental setup.

¹The source code is available at: <https://github.com/tum-lkn/NCSbench>.



Fig. 2: Time evolution of the QoS KPIs: packet loss and delays of the sensor, controller, network, round-trip, and control loop.

II. NCSBENCH DEMONSTRATION

The demonstration visualizes real-time measurements of the NCSbench platform during *benchmarking experiments* in the exemplary configuration shown in Fig. 1. As defined in [6], the result of each benchmarking experiment is a set of values used to derive *key performance indicators* (KPIs) used to analyse and understand the operation of the NCS. In the demonstration, as shown in Fig. 3 and 2, we visualize in real-time on a web-based GUI the evolution of the KPIs.

In particular, Quality of Service (QoS) and Quality of Control (QoC) KPIs are shown. The selected QoS KPIs are reliability (packet loss) and the delays measured following the timing model presented in [6](Fig. 2). In particular, as shown in Fig. 2, we select the controller and sensor processing delays, the network delay, the round-trip sensor-to-actuator delay, and the control loop delay. On the other hand, QoC describes the performance of the control system. As QoC KPIs, as shown in Fig. 3, we select the time evolution of the physical system's state, i.e. the robot's position and vertical angle, and of the controller's effort, i.e. the motors' commands. For each delay and QoC KPI, the minimum, mean, and maximum values are shown below the real-time graph. For QoC KPIs, the total absolute error is additionally shown.

A. Experimental Setup

As shown in Fig. 1, the demonstration is realized connecting the controller, the robot and an additional real-time GUI via a two-hop TCP/IP network. The first hop connects the controller and the GUI to a W-LAN Access Point (AP) via Ethernet, while the second hop connects the AP to the robot in two different configurations: wired Ethernet or wireless W-LAN. The robot runs on the Lego Mindstorms™ micro-controller and is battery powered. The controller runs on a RaspberryPi™, while the GUI is executed on a powerful multi-purpose 64-bit Linux computer.

Robot and controller communicate using an application protocol that consists of two messages: the *sensor value* and



Fig. 3: Time evolution of the QoC KPIs: motors' commands (bottom), robot's position (mid) and vertical angle (top).

the *actuation command* messages. The former is periodically sent by the robot to the controller, while the latter is sent as a response by the controller and contains the voltages to be applied at the motors to control the robot. In addition, both messages contain sequence numbers and timestamps used for delay measurements and packet loss detection. For demonstration purposes, upon receiving information from the robot, the controller additionally sends a *measurement* message to the GUI for real-time visualization. This step is required to decouple the control logic from the resource-intensive task of visualizing the real-time data.

ACKNOWLEDGMENT

This work was supported by the DFG Priority Programme 1914 Cyber-Physical Networking grant numbers KE1863/5-1, RA516/12-1, and CA595/7-1.

REFERENCES

- [1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, 2010.
- [2] K. Kim and P. R. Kumar, "Cyber-physical systems: A perspective at the centennial," *Proceedings of the IEEE Special Centennial Issue*, 2012.
- [3] R. A. Gupta and M. Chow, "Networked control system: Overview and research trends," *IEEE Transactions on Industrial Electronics*, 2010.
- [4] L. Zhang, H. Gao, and O. Kaynak, "Network-Induced Constraints in Networked Control Systems—A Survey," *IEEE Transactions on Industrial Informatics*, 2013.
- [5] X. Zhang, Q. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," *IEEE Transactions on Industrial Informatics*, 2016.
- [6] S. Zoppi, O. Ayan, F. Molinari, Z. Music, S. Gallenmüller, G. Carle, and W. Kellerer, "NCSbench: Reproducible Benchmarking Platform for Networked Control Systems," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC) (CCNC 2020)*, 2020.
- [7] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman, "Real-time control of a two-wheeled inverted pendulum mobile robot," *World Academy of Science, Engineering and Technology*, 2008.
- [8] Y. Kim, S. H. Kim, and Y. K. Kwak, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *Journal of Intelligent and Robotic Systems*, 2005.
- [9] H. G. Nguyen, J. Morrell, K. D. Mullens, A. B. Burmeister, S. Miles, N. Farrington, K. M. Thomas, and D. W. Gage, "Segway robotic mobility platform," in *Mobile Robots XVII*, 2004.