IMACS

Citation for published version (APA):
Mohamed, S., De, S., Bimpisidis, K., Nathan, V., Goswami, D., Corporaal, H., & Basten, T. (2019). IMACS: a framework for performance evaluation of image approximation in a closed-loop system. In R. Stojanovic, L. Jozwiak, B. Lutovac, & D. Jurisic (Eds.), *2019 8th Mediterranean Conference on Embedded Computing, MECO 2019 - Proceedings* Article 8760055 Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/MECO.2019.8760055

Document license:
CC BY-ND

DOI:
10.1109/MECO.2019.8760055

Document status and date:
Published: 01/06/2019

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 16. Nov. 2023

IMACS: A Framework for Performance Evaluation of Image Approximation in a Closed-loop System

Citation for published version (APA):
Mohamed, S., De, S., Bimpisidis, K., Nathan, V., Goswami, D., Corporaal, H., & Basten, T. (2019). IMACS: A Framework for Performance Evaluation of Image Approximation in a Closed-loop System. In 8th Mediterranean Conference on Embedded Computing (MECO) DOI: 10.1109/MECO.2019.8760055

DOI:
10.1109/MECO.2019.8760055

Document status and date:
Published: 01/01/2019

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 01. Aug. 2019

IMACS: A Framework for Performance Evaluation of Image Approximation in a Closed-loop System

Sajid Mohamed[1], Sayandip De[1], Konstantinos Bimpisidis[1], Vishak Nathan[1],
Dip Goswami[1], Henk Corporaal[1], Twan Basten[1,2]

[1]Electronic Systems Group, Eindhoven University of Technology, The Netherlands
[2]ESI, TNO, Eindhoven, The Netherlands
{s.mohamed, sayandip.de, d.goswami, h.corporaal, a.a.basten}@tue.nl, {k.bimpisidis, v.nathan}@student.tue.nl

Abstract—**Image Processing (IP) applications have become popular with the advent of efficient algorithms and low-cost CMOS cameras with high resolution. However, IP applications are compute-intensive, consume a lot of energy and have long processing times. Image approximation has been proposed by recent works for an energy-efficient design of these applications. It also reduces the impact of long processing times. The challenge here is that the IP applications often work as a part of bigger closed-loop control systems, e.g. advanced driver assistance system (ADAS). The impact of image approximations that tolerate certain error on these image-based control (IBC) systems is very important. However, there is a lack of tool support to evaluate the performance of such closed-loop IBC systems when the IP is approximated.**

We propose a framework - for both software-in-the-loop (SiL) and hardware-in-the-loop (HiL) simulation - for performance evaluation of image approximation on a closed-loop automotive IBC system (IMACS). Both simulation setups model the 3D environment in 3ds Max, and simulate the system dynamics, camera position and environment in V-REP. Our SiL setup simulates the system software in C++ or Matlab. Here, V-REP runs as a server and the software as a client in synchronous mode. Our HiL simulation setup runs the system software in the NVIDIA Drive PX2 platform and communicates to V-REP using application programming interfaces (APIs) for synchronous execution. We show the effectiveness of our framework using a vision-based lateral control example.

I. INTRODUCTION

Image-Based Control (IBC) systems are closed-loop feedback control systems which form the backbone of many modern applications like advanced driver assistance systems (ADAS), lane departure warning (LDW) systems, autonomous driving systems, visual navigation systems etc. A typical IBC system consists of a sensing task (T_s), which processes the output of a camera sensor, a computation task (T_c), which executes the control algorithm and an actuation task (T_a) that applies the decisions made by the controller (see Fig. 1a).

The sensing task (T_s) in a typical IBC system is composed of an image signal processing (ISP) pipeline which preprocesses the RAW image captured by the camera sensor and converts it to a compressed format. This is followed by a set of image processing (IP) algorithms designed as per the application requirements which extract features to be used by the control algorithm. Due to heavy workloads of the ISP pipeline and the IP algorithms (see Fig. 1a), the sensing task (T_s) becomes the main bottleneck in determining a faster sampling period for the controller. This impacts the control performance of the entire IBC system [1].

To improve the control performance of an IBC system, there is a need to reduce the worst-case response time (WCRT) of the sensing task (T_s). Image approximation - which trades-off quality of desired output for performance improvements - is a potential candidate to address this. It takes advantage of the fact that most IP algorithms as well as ISP stages are inherently error resilient [2, 3]. A lot of research effort has already been put into image approximation [4–6]. However, most of these approximation techniques just focus on the IP applications as a stand-alone component. *But IP applications may work as a part of bigger closed-loop IBC systems. This makes it challenging to analyze the impact of image approximation on the control performance of the entire IBC system and marks the need for a framework to easily analyze these trade-offs on closed-loop systems.*

In this work, we propose `IMACS`, a framework to easily evaluate the impact of image approximation on the performance of an IBC system. Existing ISP pipelines are designed for photography: they produce high-quality images for visual consumption. But our key observation is that these photography-oriented ISP pipelines are an overkill for IBC systems, as the control algorithms do not need the high level of visual quality produced by these pipelines. So, in `IMACS`, we provide configurable knobs to apply approximations on the ISP pipeline and/or IP so as to evaluate its trade-off on the control performance of the entire system.

The key contributions of this work are:

1) `IMACS`: A framework for performance evaluation of image approximation on IBC systems. We model the 3D environment in 3ds Max [7] and simulate the system dynamics, camera positions and environment using V-REP [8] (see Section III).
2) A software-in-the-loop (SiL) simulator which runs V-REP as a server and the software as a client in synchronous mode (see Section IV).
3) A hardware-in-the-loop (HiL) simulator which runs the software in NVIDIA Drive PX2 platform [9] and communicates to V-REP using application programming interfaces (APIs) (see Section V).

4) Evaluation of the effectiveness of `IMACS` using vision-based lateral control as case study (see Section VI).

The rest of the paper is organized as follows: Section II gives an overview of the existing full-system approximations in the literature. Proposed framework is described in Section III. Detailed description of the SiL and the HiL simulators are given in Sections IV and V respectively. Section VI provides the results for our framework and finally, Section VII concludes the paper with a glimpse on possible future research directions.

II. RELATED WORK

Approximate computing is gaining popularity due to the energy and performance benefits it offers in error resilient applications. Although its effects have been thoroughly studied at different levels across the computing stack, there is a limited amount of research on the impact of approximations on the bigger closed-loop system.

An algorithmic approximation approach for software-based applications is proposed by [10] which reduces the bit precision of the data structures. The approach proved to be quite effective, achieving orders of magnitude improvements both in computation and communication costs. An architectural approximation approach is reported in [11] which focuses on code acceleration using an approximate neural accelerator. A hardware-based approximation technique for computer vision is proposed by [3]. The paper proposes an image sensor design for computer vision tasks that skips selected ISP stages and generates inaccurate image data, achieving a considerable reduction in energy while keeping the classification accuracy of typical DNN datasets, such as the CIFAR-10, within acceptable levels. However, [10], [11] and [3] do not evaluate the impact of such approximations on high-level closed-loop systems.

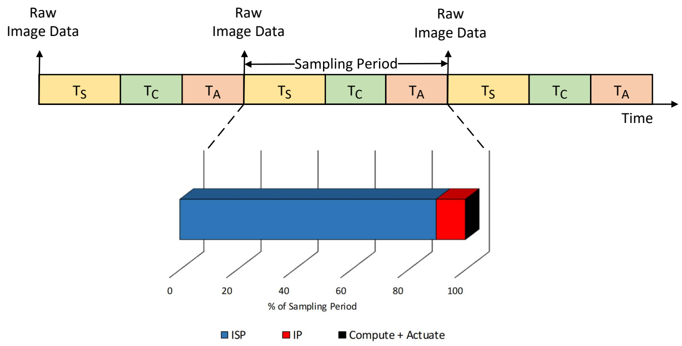
A full system approximation technique is proposed in [12]. This method approximates different stages of a smart-camera system like sensor, memory, compute and communication subsystems. Another case study, showing advantages of approximate computing in the domain of biometric security systems, targeting an iris scanning application is demonstrated in [13]. The impact of approximate computing on a high-efficiency video coding (HEVC) encoder is explored in [14] leading to considerable energy benefits that can enable the use of such encoders in ultra-low power applications such as in the Internet of Things (IoT) domain. Approximation of compute-intensive nonlinear model predictive control (NMPC) algorithms for deployment in low-power embedded systems is shown in [15]. The paper proposes a hardware-based approximation technique for the NMPC computation and assumes that full state information is available. However, image approximation that introduce errors into the state information and its effects on the control performance are not considered.

Analyzing the impact of approximating a subsystem on a closed-loop IBC system is a challenging task and requires proper tool support for applying synergistic cross-layer approximations. This challenge is addressed using our framework. To the best of our knowledge, this is the first academic framework that provides tool support for the evaluation of image approximation on closed-loop IBC systems particularly in the automotive domain.

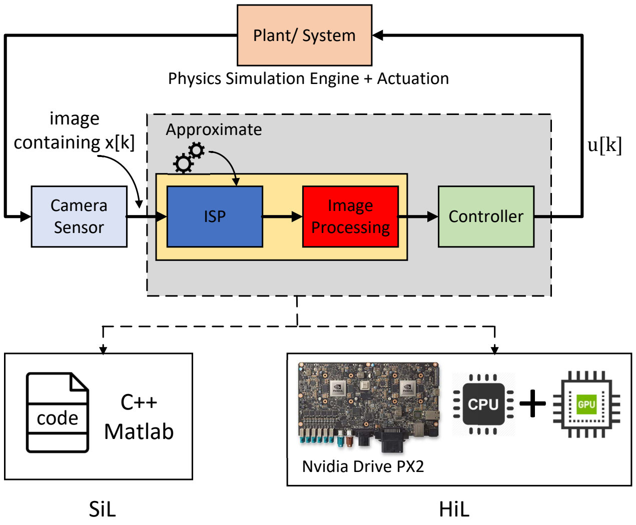
Our framework provides tool support for both SiL and HiL simulation. Existing HiL simulators [16, 17] are commercial tools. These simulators are not open-sourced and hence configuring the approximation knobs for evaluating the performance becomes difficult and often time infeasible.

III. PROPOSED FRAMEWORK: IMACS

In this work, we propose a framework `IMACS` which helps in evaluating the impact of image approximation on the performance of a closed-loop IBC system. Our framework consists of a physics simulation engine that simulates the plant or system and software that can be simulated either in a SiL or a HiL setting.



(a) Tasks in an IBC system: Runtimes for different sub-tasks are shown for a vision-based lateral control example.



(b) SiL and HiL flow for `IMACS`

Fig. 1: An overview of the proposed framework: `IMACS`

The main advantages of our `IMACS` framework are:

1) It allows profiling and characterizing the IBC loop.
2) It allows approximation of the ISP stages and IP.
3) It allows configuring the different approximation knobs and evaluating the impact on the performance of the closed-loop system.

4) It allows an in-depth analysis of the trade-offs between approximation and control performance. E.g. approximating the ISP stages reduces the sensing time and thus improves the sampling period of the controller. This results in a better control performance.

We explain `IMACS` using a vision-based lateral control example [1]. As shown in Fig. 1b, the setup consists of a vehicle with a (front-mounted) camera sensor, an ISP pipeline which preprocesses the RAW image data obtained from the camera sensor and converts to compressed data, a set of IP algorithms which identify lane markers to calculate the lateral deviation of the vehicle from the middle of the lane and a controller which sets the steering angle in order to minimize the lateral deviation. Finally, the physics simulation engine simulates the vehicle dynamics, the environment and actuates the response from the controller.

Fig. 1a shows that the entire vision-based lateral control system can be divided into three main tasks: sensing (T_s), compute (T_c) and actuate (T_a). T_s task is further composed of the ISP pipeline and IP tasks. The sampling period of the controller is the time between the start of two successive T_s tasks. The camera images contain the system state information $x[k]$, e.g. lateral deviation, and are derived by the IP algorithms. The controller then uses the system states $x[k]$ to compute the control input $u[k]$ which is the steering angle in this study.

Application profiling on an 8 core generic intel i7 processor running at 2.6 GHz shows that sensing (T_s) is the most compute intensive task and thus, the main bottleneck. Also, a closer look into T_s shows that ISP takes 89% of the total runtime of the application (see Fig. 1a). So, in `IMACS`, we provide configurable knobs to apply approximations on the ISP pipeline and easily evaluate its impact on the control performance of the entire system.

IV. SOFTWARE-IN-THE-LOOP (SIL) SIMULATOR

A SiL simulation is necessary to test the functional correctness of the developed software. Our SiL simulator simulates the plant or system using a physics simulation engine (see Fig. 1). The software in the SiL simulator takes as input the raw image containing state information $x[k]$ from the camera sensor. The raw image goes through the ISP stages as defined in our software (currently `IMACS` supports C++ and Matlab code). The ISP stages generate a compressed image that is then processed by the image processing algorithm. The image processing algorithm extracts the state information $x[k]$ and feeds it to the controller algorithm. The controller computes the input $u[k]$ and communicates it to the physics simulation engine. The frame rate for generating the raw image can either be set in the simulation engine or triggered synchronously using our simulator.

Currently, `IMACS` uses the virtual robot experimentation platform (V-REP) [8] as the physics simulation engine. V-REP is a robot simulation framework that enables the development of complex simulation scenarios, allowing the control entity of the simulation (our software) to be external. As external control entity, V-REP's remote API is used in a client-server architecture, in which the server is the V-REP simulation and the client is the control entity written in C++ or Matlab code. V-REP's synchronous communication operation is being used, which allows passing each simulation step within V-REP in full synchronization with the software. The interaction with the plant in V-REP is limited to extracting the image taken by the camera and setting the control input $u[k]$ on the vehicle simulated in V-REP.

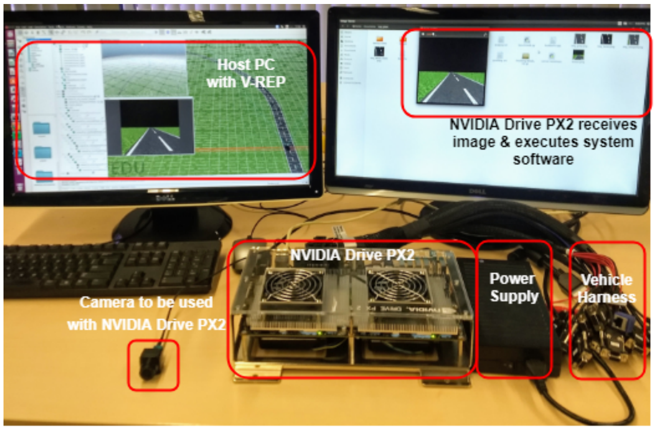


Fig. 2: HiL simulator setup

V. HARDWARE-IN-THE-LOOP (HIL) SIMULATOR

A Hardware-in-the-Loop (HiL) simulation is essential to validate functionality and ensure safe timing behaviour of a closed-loop system. HiL simulation offers realistic validation and evaluation of the developed software compared to pure offline simulation and in-vehicle rapid prototyping approaches [16]. Our HiL simulator is illustrated in Fig. 1. The software we have developed for our closed-loop system runs on the Nvidia Drive PX2 platform.

NVIDIA Drive PX2 is a scalable, open autonomous vehicle computing platform that is intended to be the "brain" of an autonomous vehicle [9]. The NVIDIA Drive PX2 comprises of 2 Tegra Systems-on-Chip (SoCs) that communicate to each other via ethernet. Each Tegra SoC has 2 CPU clusters. One cluster contains 4 ARM Cortex A57 cores and the other contains 2 NVIDIA Denver2 cores. The clusters are connected through a high-performance network interconnect. Each of the Tegra SoC also has 2 Graphical processing Units (GPUs) - an integrated Pascal GPU and a discrete GPU - with CUDA support. The integrated Pascal GPU has 256 CUDA cores and the discrete GPU has 1154 CUDA cores.

The HiL simulator setup we developed is shown in Fig. 2. Our HiL simulator interacts with the physics simulation engine V-REP using TCP/IP [18]. A synchronous mode of execution between V-REP and our Nvidia platform is achieved using the remote API. This ensures a lock-step simulation such that V-REP can wait until the control input is received from the platform. Enforcing a constant sampling period and delay necessary for the control design can be ensured by this synchronization. In addition to the advantages of the `IMACS` framework explained in Section III, our `IMACS` HiL simulator

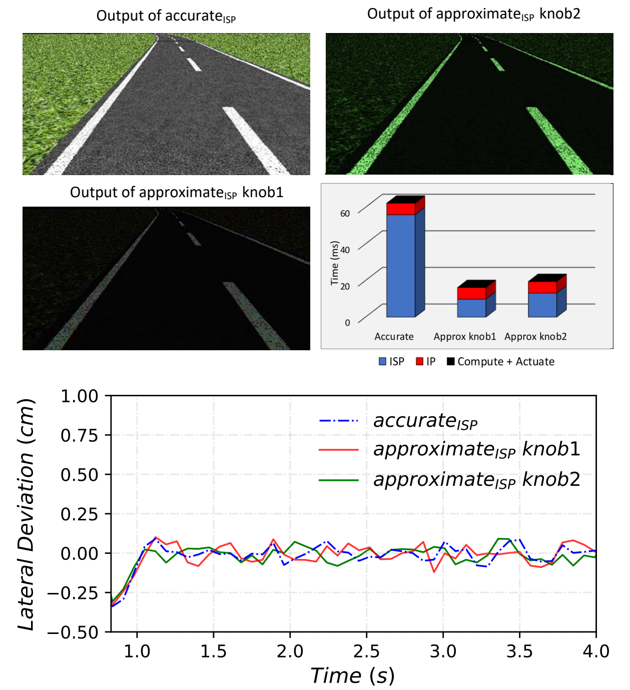


Fig. 3: `IMACS` framework results for accurate vs approximate image with respect to profiling and performance evaluation.

also lets us evaluate the trade-offs between performance of our system for different mapping options for the software tasks.

VI. Results

We illustrate our results using the vision-based lateral control example [1]. Fig. 3 shows the image output we obtain when the V-REP images are passed through an accurate ISP and then an approximate ISP with two different configuration knobs. Knob1 performs only color mapping while knob2 performs only demosaicing [3]. Using `IMACS` we profile the timing values for processing the accurate and the approximated images. We observe a 5.5x and a 4x improvement in timing for knob1 and knob2 respectively. We notice that the control performance indeed deteriorates for the approximate images. However the performance is still within acceptable bounds (worst-case lateral deviation of 6% is obtained with respect to the original). Note that Fig. 3 does not incorporate the improved timing values to get a shorter sampling period, which may in turn improve the control performance. An extensive trade-offs analysis considering all these cases is planned for future work.

VII. Conclusion

We present the `IMACS` framework for performance evaluation of image approximation in a closed-loop image-based control (IBC) system. We integrate both software-in-the-loop (SiL) and hardware-in-the-loop (HiL) simulators in `IMACS`. `IMACS` uses V-REP as the physics simulation engine for simulating the plant/system. `IMACS` currently supports C++ and Matlab codes for SiL and the Nvidia Drive PX2 platform for HiL simulations.

An extensive trade-offs analysis between image approximation and control performance is planned for future work. We plan to compare different algorithmic approximation techniques and their effect on the control performance using `IMACS`. The `IMACS` framework is available for download from http://www.es.ele.tue.nl/cps/automotive/#imacs.

VIII. Acknowledgment

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875 (oCPS).

References

[1] S. Mohamed, D. Zhu, D. Goswami, and T. Basten, "Optimising quality-of-control for data-intensive multiprocessor image-based control systems considering workload variations," in *DSD*, 2018, pp. 320–327.

[2] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *DAC*, 2013.

[3] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in *ICCV*, 2017.

[4] S. Mittal, "A Survey of Techniques for Approximate Computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–33, 2016.

[5] F. S. Snigdha, D. Sengupta, J. Hu, and S. S. Sapatnekar, "Optimal design of jpeg hardware under the approximate computing paradigm," in *DAC*, 2016, pp. 1–6.

[6] W. El-Harouni *et al.*, "Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding," in *DATE*, 2017, pp. 1384–1389.

[7] K. L. Murdock, *3ds Max 2012 bible*. John Wiley & Sons, 2011, vol. 783.

[8] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *IROS*, 2013.

[9] "Nvidia drive, scalable ai platform for autonomous driving." [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-platform/

[10] C. Chen, J. Choi, K. Gopalakrishnan, V. Srinivasan, and S. Venkataramani, "Exploiting approximate computing for deep learning acceleration," in *DATE*, 2018.

[11] S. De, J. Huisken, and H. Corporaal, "Designing energy efficient approximate multipliers for neural acceleration," in *DSD*, 2018.

[12] A. Raha and V. Raghunathan, "Approximating beyond the processor: Exploring full-system energy-accuracy tradeoffs in a smart camera system," *IEEE Trans. on VLSI Systems*, vol. 26, no. 12, pp. 2884–2897, 2018.

[13] S. Hashemi, H. Tann, F. Buttafuoco, and S. Reda, "Approximate computing for biometric security systems: A case study on iris scanning," in *DATE*, 2018.

[14] A. Mercat, J. Bonnot, M. Pelcat, W. Hamidouche, and D. Menard, "Exploiting computation skip to reduce energy consumption by approximate computing, an hevc encoder case study," in *DATE*, 2017, pp. 494–499.

[15] A. Raha, A. Chakrabarty, V. Raghunathan, and G. T. Buzzard, "Embedding approximate nonlinear model predictive control at ultrahigh speed and extremely low power," *IEEE Trans. on Control Systems Technology*, pp. 1–8, 2019.

[16] W. Deng, Y. H. Lee, and A. Zhao, "Hardware-in-the-loop simulation for autonomous driving," in *Annual Conference of Industrial Electronics*, 2008.

[17] "Nvidia drive constellation: Virtual reality autonomous vehicle simulator." [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-constellation/

[18] B. A. Forouzan, *TCP/IP protocol suite*. McGraw-Hill, Inc., 2002.