Latency prediction in 5G for control with deadtime compensation

Ruuskanen, Johan; Peng, Haorui; Martins, Alexandre

Published in:
IoT-Fog '19 Proceedings of the Workshop on Fog Computing and the IoT

2019

Document Version:
Publisher's PDF, also known as Version of record

Latency Prediction in 5G for Control with Deadtime Compensation

Johan Ruuskanen
johan.ruuskanen@control.lth.se
Department of Automatic Control,
Lund University

Haorui Peng
haorui.peng@eit.lth.se
Department of Electrical and
Information Technology,
Lund University

Alexandre Martins
alexandre.martins@axis.com
Axis Communications &
Department of Automatic Control,
Lund University

ABSTRACT

With the promise of increased responsiveness and robustness of the emerging 5G technology, it is suddenly becoming feasible to deploy latency-sensitive control systems over the cloud via a mobile network. Even though 5G is herald to give lower latency and jitter than current mobile networks, the effect of the delay would still be non-negligible for certain applications.

In this paper we explore and demonstrate the possibility of compensating for the unknown and time-varying latency introduced by a 5G mobile network for control of a latency-sensitive plant. We show that the latency from a prototype 5G test bed lacks significant short-term correlation, making accurate latency prediction a difficult task. Further, because of the unknown and time-varying latency our used simple interpolation-based model experiences some troubling theoretical properties, limiting its usability in real world environments. Despite this, we give a demonstration of the strategy which seems to increase robustness in a simulated plant.

CCS CONCEPTS

• **Networks** → **Network performance evaluation**; *Wireless access points, base stations and infrastructure*; • **Computer systems organization** → *Embedded and cyber-physical systems*;

KEYWORDS

Latency prediction, 5G, Deadtime compensation, Time varying delay

ACM Reference Format:
Johan Ruuskanen, Haorui Peng, and Alexandre Martins. 2019. Latency Prediction in 5G for Control with Deadtime Compensation. In *IoT-Fog '19: Workshop on Fog Computing and the IoT, April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3313150.3313227

1 INTRODUCTION

In IoT scenarios there are many potential benefits of extracting control and decision algorithms from individual agents or plants and placing them in the cloud. Such benefits include increased available computational power, decreased cost through the economies of scale effect as each plant does not have to carry expensive computational hardware, and better performance due to absolute awareness between competing agents in the same area.

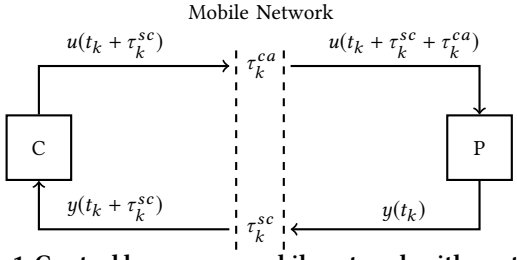

Figure 1: Control loop over a mobile network with controller C and plant P. The measurement at time t_k is denoted as $y(t_k)$, while the control signal is denoted as $u(t_k)$. The total delay introduced by the mobile network can be split as $\tau_k = \tau_k^{sc} + \tau_k^{ca}$ where τ_k^{sc} is the delay between plant-controller and τ_k^{ca} between controller-plant.

With the advent of Ultra-Reliable and Low-Latency Communications (URLLC) 5G [9] there is now a possibility to control latency-sensitive applications across mobile networks as demonstrated by Skarin & Tärneberg et al. [10], greatly increasing the flexibility of IoT solutions in real-world scenarios.

For latency-sensitive or mission-critical control applications, it is of high importance that the delay between the plant and the controller remains low. By performing *deadtime compensation* (DTC) [7] the effects of the latency τ could in theory be mitigated. The DTC can be performed by predicting the plant measurement τ time units into the future, and letting the controller act on this predicted measurement instead.

Over a mobile network the latency will vary with time, and the entire delay at current cycle k is unfortunately not measurable from the controllers perspective. This can be seen by considering Figure 1, the delay τ_k^{ca} between the controller and the plant has simply not occurred yet. The latency must then also be predicted *before* predicting the plant measurement, to know how far into the future to look.

While control systems over networks and wireless communications has been extensively studied [3, 8], the advent of 5G creates new possibilities and challenges for such systems. Although the case of DTC with latency prediction for control of latency-sensitive plants under time-variable network delays has been studied in e.g. [13], the effects from a real 5G link in such systems remains to the best of our knowledge uncharted.

In this workshop paper we thus aim at examining ways to perform latency prediction for DTC on a 5G link and to demonstrate

IoT-Fog '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6698-4/19/04…$15.00
https://doi.org/10.1145/3313150.3313227

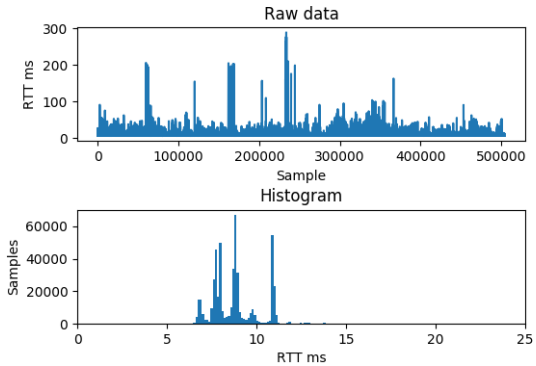

Figure 2: The raw data and histogram from the RTT through the LuMaMi. The histogram has been cropped at 25 ms.

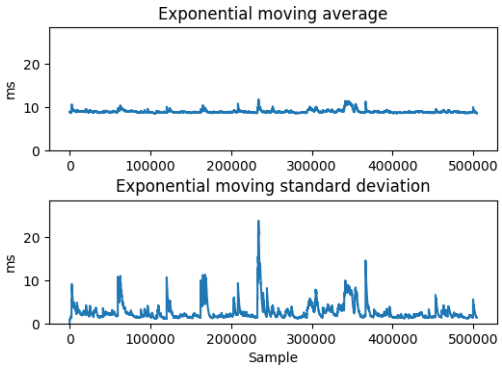

Figure 3: The exponential moving expected value and standard deviation, calculated with a weight of 0.001**.**

its usability by extending the environment created by Skarin & Tärneberg et al. [10]. We evaluate latency data gathered from a prototype 5G base station and compare suitable latency predictors. Using these we examine the suitability of a simple, interpolation-based model for DTC when the delays are time-varying. Finally, using our implementation we demonstrate a working controller using DTC with latency prediction on a simulated, latency-sensitive plant, the Ball-and-Beam process [11].

2 CHARACTERISTICS OF SYSTEM LATENCY

The Lund Massive MIMO (LuMaMi) antenna system [6] was used as a prototype 5G base station to capture and examine transmission latencies. Data was gathered using the well-known command `ping` between two computers connected via the 5G link. A sampling time of 50ms was used to simulate packages sent at a frequency of 20Hz. The latencies and its histogram are presented in Figure 2 as the *round trip time* (RTT) in milliseconds. As can be seen the LuMaMi has a tendency to generate latency spikes above 50ms.

The latency mean and variance are assumed to be time-varying, or non-stationary and estimated using the exponential weighted moving average/variance (EWMA/V) filter [2]. The results are shown in Figure 3. As can be seen, the mean and variance seems to at least have a stationary baseline.

We further explore the correlation between latency measurements, also known as the autocorrelation [4], to see whether there is structure in the signal that can be used to predict further events. As with the mean/variance the autocorrelation is assumed to be non-stationary and therefor calculated in small, overlapping bins. The result can be seen in Figure 4. Here it is clear that the autocorrelation is non-stationary and for most part lacks significant correlation. This indicates that it will be difficult to find a predictor that outperforms a mean value filter.

Based on these results, two latency predictors was considered. The first is based on our EWMA, the predictor is formed by guessing on the current estimated mean which could arguably provide a competitive prediction as the correlation is low.

The second predictor considered is the *auto-regressive moving-average* ARMA(p,q) [4] filter, which models the signal output y_k at time k as a linear combination of previous measurements $\{y_{k-1}, \ldots, y_{k-p}\}$ and i.i.d. zero-mean Gaussian noise $\{e_k, \ldots, e_{k-p}\}$,

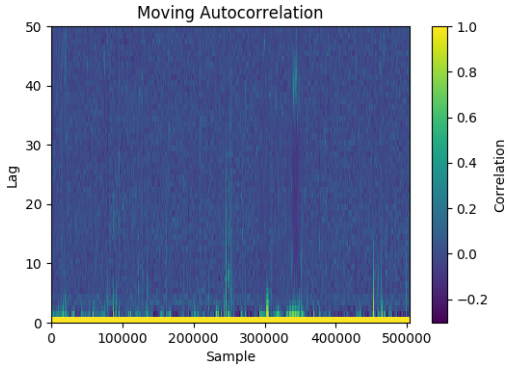

Figure 4: The autocorrelation for 50 lags computed over bins of 500 samples, overlapping by 80%.

i.e.

$$y_k + a_1 y_{k-1} + \cdots + a_p y_{k-p} = e_k + c_1 e_{k-1} + \cdots + c_q e_{k-q}.$$

The coefficients $A = [a_1, \ldots, a_p], C = [c_1, \ldots, c_q]$ can then be fitted such that the model output and data are as similar as possible.

However, since the latency exhibits non-stationary behavior, it is unlikely that the optimal coefficients A, C will be the same for all time instances. Instead A, C can be fitted on-line using the Kalman filter [5] with the following state space model [4]

$$x_{k+1} = F_k x_k + w_k \qquad w_k \sim \mathcal{N}(0, Q),$$

$$y_k^{(m)} = H_k x_k + v_k \qquad v_k \sim \mathcal{N}(0, R),$$

where the states $x_k = [A_k,\ C_k]^T$, the measurement matrix $H_k = [-y_{k-1}^{(m)}, \ldots, -y_{k-p}^{(m)}, \hat{e}_{k-1}, \ldots, \hat{e}_{k-q}]$, the mean reduced measurements $y_k^{(m)} = y_k - \hat{\mu}_k$, and the transition matrix $F_k = I$. The estimated residual e_k can be retrieved as $\hat{e}_k = y_k^{(m)} - H_k \hat{x}_k$ and the estimated mean μ_k from the EWMA filter. The choice of F_k as the identity matrix is based on the fact that we assume no dynamics of the coefficients A, C but let their values be solely dependent on the measurements. By experimental tuning the parameters where chosen as $p = q = 2$, $Q = 1\text{e}{-6} \cdot I$ and R_k equal to the variance estimated by the EWMV.

Both latency predictors were evaluated on the data. The residuals $\hat{e}_k = y_k - \hat{y}_k$ are displayed in Figure 5 and the ARMA coefficients in Figure 6. Here it is clear that the EWMA and ARMA performs roughly equal, as the histograms are similar. However, by considering the absolute residuals $|r_k|$ the ARMA filter seems to be

a little better at capturing outliers. The *root mean squared error* (RMSE) of the ARMA filter was ≈ 2.8 and for the EWMA ≈ 3.4. The performance similarity is further enhanced by the fact that the ARMA-polynomials shown in Figure 6 are often situated around 0, implying that only little information for inference can be extracted.

3 DTC USING INTERPOLATION

As a crude model of the plant, different interpolation design are considered. We generate state space representations of the interpolation models by discretizing the n-th order continuous integrator using the *zero-order hold* (ZOH) method [12]. The state space for the n-th order discretized integrator becomes

$$F(h_k))^{(n)} = \begin{pmatrix} 1 & h_k & \frac{h_k^2}{2} & \cdots & \frac{h_k^n}{n!} \\ 0 & 1 & h_k & \cdots & \frac{h_k^{n-1}}{(n-1)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

$$H^{(n)} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$x_k^{(n)} = \begin{pmatrix} y_k & \frac{d}{dt}y_k & \frac{d^2}{dt^2}y_k & \cdots & \frac{d^n}{dt^n}y_k, \end{pmatrix}$$

where h_k is the time step. Using the Kalman filter [5] the states x_k can be tracked. At each new measurement that arrives, the states are propagated one step with $h_k = t_k^m - t_{k-1}^m$ where t_k^m is the time at measurement event k. When the controller is triggered, the measurement-prediction can then be calculated using $h_k = \hat{\tau}_k + (t_k - t_+^m)$ as

$$\hat{y}_{k+1} = HF(h_k)x_k = y_k + h_k\dot{y}_k + \frac{h_k^2}{2}\ddot{y}_k + \cdots$$

where $\hat{\tau}_k$ is the predicted latency from the precious section, t_k the current time and t_+^m is the most recent measurement event.

3.1 Problems from layered predictors

A predicted latency will affect the quality of the measurement prediction, compared to if τ_k was known. Here we will examine what effects this might have for our interpolation model by comparing the error between two measurement predictors $\hat{y}_{k+1}^a, \hat{y}_{k+1}^b$ with known and unknown τ_k respectively. For simplicity, we can let τ_k denote the entire h_k in this subsection. Assume that the latency prediction is unbiased, i.e. $\mathbb{E}(\hat{\tau}_k) = \tau_k$ and has a variance of $\mathbb{V}(\hat{\tau}_k) = \sigma_k^2$. The expected value and variance of the error then becomes

$$\begin{aligned} &\mathbb{E}(\hat{y}_{k+1}^a - \hat{y}_{k+1}^b) \\ &\quad = \mathbb{E}\left(y_k + \tau_k\dot{y}_k + \frac{\tau_k^2}{2}\ddot{y}_k - y_k - \hat{\tau}_k\dot{y}_k - \frac{\hat{\tau}_k^2}{2}\ddot{y}_k + O(\dddot{y}_k)\right) \\ &\quad = 0 + \left(\tau_k^2 - \mathbb{E}\left(\hat{\tau}_k^2\right)\right)\frac{\ddot{y}_k}{2} + \mathbb{E}\left(O(\dddot{y}_k)\right); \\ &\mathbb{V}(\hat{y}_{k+1}^a - \hat{y}_{k+1}^b) \\ &\quad = \mathbb{V}(\hat{y}_{k+1}^a) - 2\mathbb{C}(\hat{y}_{k+1}^a, \hat{y}_{k+1}^b) + \mathbb{V}(\hat{y}_{k+1}^b) \\ &\quad = 0 - 2\cdot 0 + \mathbb{V}\left(y_k + \hat{\tau}_k\dot{y}_k + \frac{\hat{\tau}_k^2}{2}\ddot{y}_k\right) \\ &\quad = \sigma_k^2\dot{y}_k^2 + 2\mathbb{C}\left(\hat{\tau}_k y_k, \frac{\hat{\tau}_k^2}{2}\ddot{y}_k + O(\dddot{y}_k)\right) + \mathbb{V}\left(\frac{\hat{\tau}_k^2}{2}\ddot{y}_k + O(\dddot{y}_k)\right). \end{aligned}$$

We see that the error between the two predictors is unbiased only if $n \leq 2$, for higher order models to be unbiased then either (i) the difference between second moment and squared expected value must be equal and this is only true if the variance is zero or (ii) all higher order terms must cancel out which is highly unlikely.

Further, for all model orders $n \geq 2$ the variance of the error is inherently dependent on the variance of our predicted latency, and scaled with $\dot{y}_k$. Thus the spread and quality of the prediction will be volatile, depending on the currents state of the plant.

```
Init c = 0, t_old = now(), D_list = empty sorted list;
switch on new event do
    case interrupt at incomming packet p_c at time t do
        d := d − dt ∀d ∈ D_list, dt = t − t_old;
        D_list.append({d_c, p_c}), d_c = data[c]/2;
        wait(min d ∈ D_list), t_old = t, c := c + 1;
    end
    case triggered wait() command at time t do
        d := d − dt ∀d ∈ D_list, dt = t − t_old;
        {d_c, p_c} = D_list.pop();
        wait(min d ∈ D_list), t_old = t;
        return p_c
    end
end
```

Algorithm 1: LuMaMi delay emulator.

4 TEST ENVIRONMENT

To demonstrate the approach, we extended the environment implemented by Skarin & Tärneberg et al. [10] written in Calvin [1], an IoT framework developed at Ericsson Research. Two new Calvin functions, or actors, were created; a delay emulator to streamline development and testing and a modified PID controller that incorporates both latency and measurement prediction.

As a demonstrative system, we set up two cascaded modified PID controllers tasked with controlling a simulated Ball-and-Beam plant sampled at 20Hz. The packet transmissions between the plant and the controllers were delayed both ways using the emulated LuMaMi link.

4.1 LuMaMi delay emulator

As the LuMaMi is a complicated piece of machinery, the process from start-up to a working antenna is slow and cumbersome. To avoid repeating the startup procedure, a new Calvin actor was created that enables *emulating* the delay instead using the previously gathered latency data from the LuMaMi.

This emulation strategy shown in Algorithm 1 will keep the inter-sample dependencies from the LuMaMi and enable the delay to give rise to unordered samples. The retrieved delay is divided by 2 to emulate a one-way delay. This will unfortunately reduce the jitter in our emulated delay with a small amount as we are essentially averaging over the two delays τ^{sc} and τ^{ca}. This loss was deemed acceptable.

4.2 Modified PID controller

Each PID controller was fitted with both latency and measurement predictors. The latency was predicted using the EWMA filter, as its performance was comparable to the ARMA filter but computationally cheaper. As measurement predictor the interpolation model

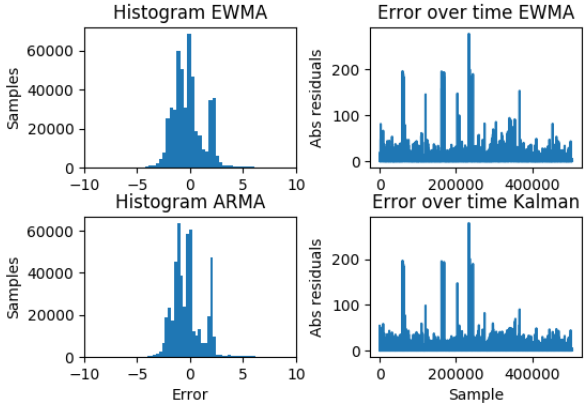

Figure 5: Analysis of the prediction residuals generated by the EWMA and ARMA filters. The histogram has been truncated between $[-10, 10]$, as no visible residuals are situated above this limit.

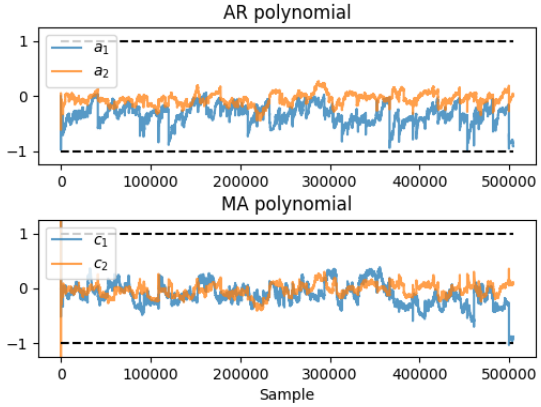

Figure 6: Analysis of the constants for the adaptive ARMA polynomials.

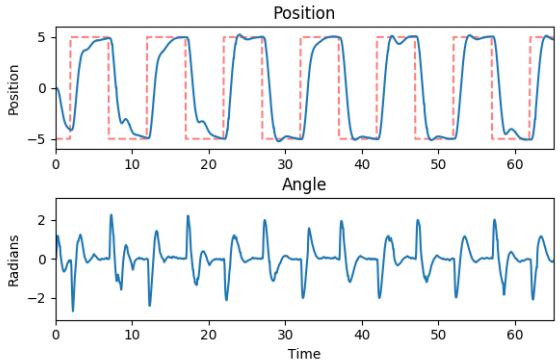

Figure 7: Controlling the plant over emulated 5G without any DTC.

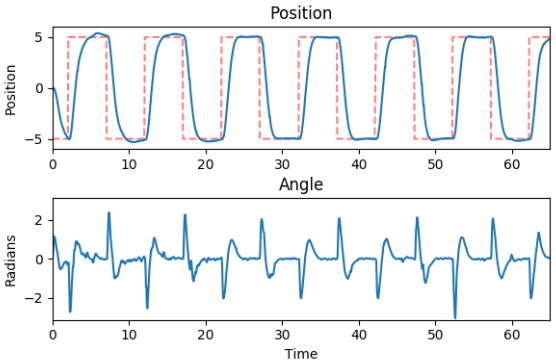

Figure 8: Controlling the plant over emulated 5G with predicted latency and interpolation-based DTC.

using $n = 2$ was considered as it is the highest order model whose gained error from the predicted latency is unbiased.

The actual latency τ_{k-1} between a controller and the plant can be measured by letting the plant return a confirmation packet to the controller at each received control signal packet.

At each new measurement arrival, the model states are updated, τ_k and y_{k+1} predicted, and a control signal u_k calculated and transmitted. When no new measurement packet has been received in the last $h_c = 50$ms, the prediction and control signal calculation is simply performed for the delay $\hat{\tau}_k + h_c$ instead. The covariance matrices of the Kalman filters were experimentally tuned to $Q = 100 \cdot I$ and $R = 0.001$ for both controllers.

4.3 Results

An experiment was conducted by running the system for 60 seconds where the reference position of the ball was changed every 5 seconds between $[-5, 5]$. The results without DTC is show in Figure 7 and with DTC in Figure 8. As can be seen the system is stable without any compensation, but the added DTC gives a smoother transition between the reference levels, indicating increased robustness.

5 CONCLUSION

In this paper we have examined and demonstrated deadtime compensation with latency prediction for a controller/plant separated by a 5G mobile network.

Since the correlation between consecutive latencies is small, the performance of the two examined latency predictors is similar. From the predicted latency, deadtime compensation was performed using a simple interpolation-based model. Our demonstration suggests that even our crude plant model provides some improvement.

Our small experiment does in no manner present any conclusive evidence, but warrants future research. A deeper and wider experimental comparison with the real process over the real LuMaMi with multiple connected users is therefor a clear next step. Whether or not the low correlation is platform specific is worth considering. Further, any correlation effects of having multiple users connected through the LuMaMi has not been tested and is of high interest. If increasing strain adds not only latency/jitter, but also correlation then the choice of latency predictor would suddenly play a larger role in real settings. Finally, comparison between the interpolation model and standard DTC methods in this setting should be performed. An interesting question is how the predicted latency would affect other choices of plant models.

ACKNOWLEDGMENTS

This work was partially supported by Nordforsk Nordic Hub on Industrial IoT (HI2OT), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

REFERENCES

[1] Ericsson. 2015. Calvin. https://github.com/EricssonResearch/calvin-base.
[2] Tony Finch. 2009. Incremental calculation of weighted mean and variance. University of Cambridge Computing Service. http://people.ds.cam.ac.uk/fanf2/hermes/doc/antiforgery/stats.pdf
[3] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. 2007. A Survey of Recent Results in Networked Control Systems. *Proc. IEEE* 95, 1 (Jan 2007), 138–162. https://doi.org/10.1109/JPROC.2006.887288
[4] Andreas Jakobsson. 2015. *An Introduction to Time Series Modeling*. Studentlitteratur AB.
[5] Rudolf E. Kálmán. 1960. A new approach to linear filtering and prediction problems. *Transactions of the AMSE-Journal of Basic Engineering* Series D (1960), 35–45.
[6] Steffen Malkowsky, Joao Vieira, Liang Liu, Paul Harris, Karl Nieman, Nikhil Kundargi, Ian C. Wong, Fredrik Tufvesson, Viktor Öwall, and Ove Edfors. 2017. The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation. *IEEE Access* 5 (2017), 9073–9088. https://doi.org/10.1109/access.2017.2705561
[7] Julio E. Normey-Rico and Eduardo. F. Camacho. 2008. Dead-time compensators: A survey. *Control Engineering Practice* 16, 4 (apr 2008), 407–428. https://doi.org/10.1016/j.conengprac.2007.05.006
[8] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson. 2018. Wireless Network Design for Control Systems: A Survey. *IEEE Communications Surveys Tutorials* 20, 2 (Second quarter 2018), 978–1013. https://doi.org/10.1109/COMST.2017.2780114
[9] Mansoor Shafi, Andreas F. Molisch, Peter J. Smith, Thomas Haustein, Peiying Zhu, Prasan De Silva, Fredrik Tufvesson, Anass Benjebbour, and Gerhard Wunder. 2017. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal on Selected Areas in Communications* 35, 6 (jun 2017), 1201–1221. https://doi.org/10.1109/jsac.2017.2692307
[10] Per Skarin, William Tärneberg, Karl-Erik Årzen, and Maria Kihl. 2018. Towards Mission-Critical Control at the Edge and Over 5G. In *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE. https://doi.org/10.1109/edge.2018.00014
[11] Marta Virseda. 2004. Modeling and Control of the Ball and Beam Process. Lund University. Student Paper.
[12] B. Wittenmark, K-J. Åström, and K-E. Årzén. 2016. Computer Control: An Overview. Educational Version. IFAC Proffesional Brief.
[13] H. Yoshida, T. Kumagai, and K. Satoda. 2018. Dynamic state-predictive control for a remote control system with large delay fluctuation. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*. 1–6. https://doi.org/10.1109/ICCE.2018.8326072