LQG-optimal PI and PID control as benchmarks for event-based control

Cervin, Anton

Published in:
2016 2nd International Conference on Event-Based Control, Communication, and Signal Processing, EBCCSP 2016 - Proceedings

DOI:
10.1109/EBCCSP.2016.7605255

2016

Document Version:
Peer reviewed version (aka post-print)

LQG-Optimal PI and PID Control As Benchmarks for Event-Based Control

Anton Cervin
Department of Automatic Control LTH
Lund University
Lund, Sweden
Email: anton@control.lth.se

***Abstract*—We formulate two simple benchmark problems for event-based control, where the optimal solutions in the continuous-time setting turn out to be ordinary PI and PID controllers. The benchmarks can be used to compare the performance of continuous-time, discrete-time, and various event-based controllers with regard to for instance disturbance attenuation, control effort, and average sampling or actuation rates. They can also be used to evaluate heuristic event-based PI(D) controllers and see how their performance compare to each other and to regular sampled-data control. We give two benchmark examples, where we study the trade-off between event frequency and regulator performance for a number of previously proposed approaches to event-based control.**

I. INTRODUCTION

Event-based feedback control is an old idea, but there has been a strong renewed interest following the 14th World Congress of IFAC in 1999, where two quite different papers on event-based control were presented. Åström & Bernhardsson's paper [1] contained a theoretical analysis and comparison of periodic and event-based control of first-order linear stochastic systems. Årzén's paper [2] proposed a simple, heuristically derived event-based PID controller, which was evaluated in simulations on a double-tank process.

Ever since these two papers were published, research on event-based control has been conducted along two more or less separate lines. On one hand, there has been theoretical development in event-based estimation and control, e.g., [3]–[9], where focus has been on deriving optimal solutions or solutions with guaranteed performance bounds. On the other hand, the development of more practical and industry-oriented event-based control schemes—including PID control—has also continued, e.g., [10]–[15].

In this paper, we make an attempt to bridge the gap between theory and practice by proposing two simple benchmarks for event-based control. The benchmarks are simple in the sense that they are defined for low-order linear processes and that the optimal continuous-time solutions can be interpreted as ordinary PI and PID controllers. At the same time, introducing one or more event-based elements in the control loop, the optimal solutions are often (so far) unknown, or known but computationally intractable. Because of the theoretical difficulties involved, various heuristic or suboptimal approaches have been proposed. For surveys of recent results in both optimal and suboptimal/heuristic event-based control and state estimation, see [16], [17].

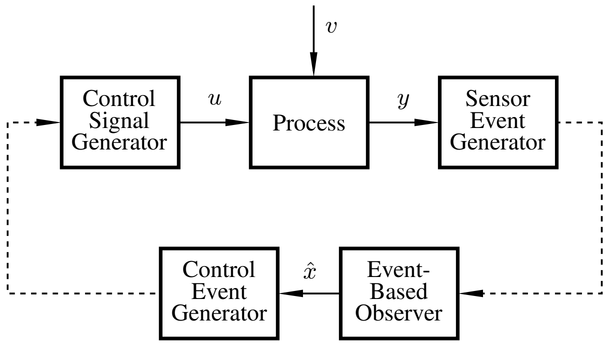

Fig. 1. A general event-based controller structure, cf. [18]. Solid lines represent continuous signal transmission, while dashed lines represent event-based transmission.

A quite general event-based controller structure for a single-input–single-output process is shown in Figure 1. The process is driven by disturbances v that should be countered by the controller. The sensor event generator and the control signal generator determine, respectively, when and how the process output y is sampled and how the process output u is updated. The sensor event generator could implement, for instance, continuous measurements, periodic sampling, quantized sampling, send-on-delta sampling, or stochastic sampling, The control signal generator could be an impulse generator, a zero-order hold, a linear system control generator, or something else. The control event generator determines when new information should be communicated to the control signal generator and could implement, for instance, continuous or periodic communication, a deadband, or a self-triggered scheme. Finally, the event-based observer transforms the intermittent information received from the sensor event generator into a state estimate $\hat{x}$ that can be used for feedback.

The proposed benchmarks in this paper are based on linear-quadratic-Gaussian (LQG) theory and make it possible to compare the performance of continuous-time, discrete-time and event-based controllers within a unified framework. The problem formulations are designed so that ordinary PI and PID controllers are optimal in the continuous-time setting if there is no cost on sensor or control events. This makes it possible to evaluate and compare different event-based PI and PID control schemes that have been proposed in the literature, both against each other and against regular sampled-data control or more advanced event-based estimation and control schemes.

Every benchmark has its limitations and the proposed benchmarks are of course no exception to this rule. The design criterion is non-standard for PID control [19], focusing on disturbance attenuation and control signal activity and disregarding robustness constraints. The disturbance model includes an integrator but is driven by white noise and hence puts more focus on high-frequency behavior than standard performance criteria for PID control that typically measure, e.g., the integral absolute error in response to a constant load disturbance. Also, the servo problem is ignored. Despite of these shortcomings, we believe that the benchmarks can still be useful for pointing out some strengths and weaknesses of the heuristic event-based PID schemes and also for posing new design challenges for optimal event-based control.

The remainder of this paper is outlined as follows. The benchmarks are detailed in Section II, where the optimal continuous-time and sampled-data solutions are given. Section III contains an example evaluation based on the optimal PI control problem, with focus on different sampling techniques and the trade-off between regulator performance and the rate of sensor events. Section IV contains another example, based on optimal PID control, with focus on the trade-off between regulator cost and the rate of control events under ZOH control. The paper is finished with some discussion and conclusions in Section V.

II. The Benchmarks

A. Benchmark Design

The benchmarks should be designed to make it possible to compare continuous-time, sampled-data, and event-based control within the same framework. It is thus natural to state the process models and the performance criteria in continuous time. The models and the criteria can then be sampled for controller design and evaluation. For event-based control schemes, it should be possible to numerically evaluate the regulator performance and the average event rates using Monte Carlo simulations of the closed-loop system.

Using LQG problems formulations we can obtain optimal linear controllers, and by choosing appropriate plant and noise models, we can derive LQG-optimal PI and PID controllers. To achieve finite gain in the controllers, we must either introduce some measurement noise in the models or penalize the control signal activity. We adopt the latter approach, since continuous-time measurement noise leads to some technical difficulties in the optimal sampled-data control design (essentially invoking the need to design an optimal anti-alias filter, see [20]).

Given a model with an integral disturbance at the process input, it is not possible to directly penalize the control signal u since it is unbounded under PI(D) control. Neither is it possible to penalize $\dot{u}$, since the control signal derivative is unbounded under sampled-data control with zero-order hold. We avoid this difficulty by penalizing the sum of the control signal and the input disturbance, which is well-defined under all control strategies and will remain bounded if the closed-loop system is stable.

Explicit robustness constraints are not included in the design problem, but the level of robustness can be adjusted by putting more or less penalty on the control signal activity, resulting in a less or more aggressive controller. Further, the benchmarks offer the possibility of experimentally evaluating the gain and delay margins of the system.

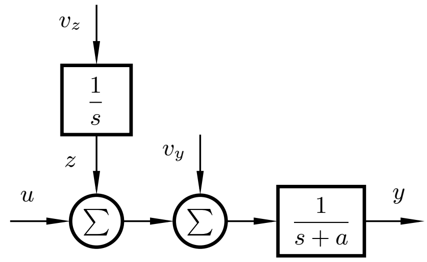

Fig. 2. First-order process with integral input disturbance.

We are now ready to formulate the LQG-optimal PI and PID control design problems.

B. LQG-Optimal PI Control

Consider the system in Figure 2, where a is a scalar parameter and v_z and v_y are independent continuous-time white noise processes with intensities r_z and 1, respectively. The control objective is to minimize the continuous-time cost function

$$J = \mathrm{E}\left\{ q_y y^2 + (u{+}z)^2 \right\} \tag{1}$$

where q_y is a scalar weight. To have a well-defined problem, we assume that $r_z, q_y > 0$.

As usual, in LQG design we may separate the problem into optimal state feedback and optimal state estimation. The optimal state feedback, assuming that a state estimate $\hat{z}$ is available, is

$$u = -l_y y - \hat{z} \tag{2}$$

where l_y is obtained by solving the associated algebraic Riccati equation, yielding

$$l_y = \sqrt{a^2 + q_y} - a \tag{3}$$

A Kalman filter for z can be designed by assuming (just for now) that $\dot{y}$ may be used in the observer. The reduced-order Kalman filter is then given by

$$\dot{\hat{z}} = k_z(\dot{y} + ay - \hat{z} - u) \tag{4}$$

where k_z is obtained by solving the associated algebraic Riccati equation, yielding

$$k_z = \sqrt{r_z} \tag{5}$$

Combining (2) and (4), and solving for u, the complete controller can be written in input-output form as

$$U(s) = -\frac{(l_y + k_z)s + k_z(l_y + a)}{s} Y(s) \tag{6}$$

Comparing this with a standard PI controller,

$$U(s) = -K\left(1 + \frac{1}{sT_i}\right) Y(s) \tag{7}$$

we can identify the PI controller parameters as

$$\begin{aligned} K &= l_y + k_z \\ T_i &= \frac{l_y + k_z}{k_z(l_y + a)} \end{aligned} \tag{8}$$

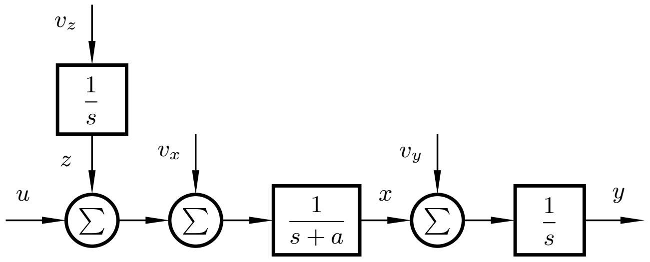

Fig. 3. Second-order process with integral input disturbance.

We note that, for a given process parameter a, there is a direct correspondence between the LQG design weights q_y and r_z and the PI parameters K and T_i.

C. LQG-Optimal PID Control

Consider the system in Figure 3, where a is a scalar parameter and v_z, v_x, and v_y are independent continuous-time white noise processes with intensities r_z, r_x, and 1, respectively. The control objective is now to minimize the continuous-time cost function

$$J = \mathrm{E}\left\{ q_y y^2 + q_x x^2 + (u+z)^2 \right\} \tag{9}$$

where q_y and q_x are scalar weights. To have a well-defined problem, we assume that $r_z, q_y > 0$ and $q_x, r_x \geq 0$.

Proceeding similarly to above, we first assume that state estimates $\hat{x}$ and $\hat{z}$ are available and design a state feedback law

$$u = -l_y y - l_x \hat{x} - \hat{z} \tag{10}$$

where the optimal gains are given by

$$\begin{aligned} l_y &= \sqrt{q_y} \\ l_x &= \sqrt{a^2 + 2\sqrt{q_y} + q_x} - a \end{aligned} \tag{11}$$

Then, again assuming that $\dot{y}$ may be used in the observer, a reduced-order Kalman filter is given by

$$\begin{aligned} \dot{\hat{z}} &= k_z(\dot{y} - \hat{z}) \\ \dot{\hat{x}} &= a\hat{x} + \hat{z} + u + k_x(\dot{y} - \hat{z}) \end{aligned} \tag{12}$$

with the optimal gains

$$\begin{aligned} k_z &= \sqrt{r_z} \\ k_x &= \sqrt{a^2 + 2\sqrt{r_z} + r_x} - a \end{aligned} \tag{13}$$

Combining (10) and (12) and solving for u, the complete controller can be written in input-output form as

$$U(s) = -\frac{(k_z + l_y + k_x l_x)s^2 + (l_y(k_x + a) + k_z(l_x + a))s + k_z l_y}{s^2 + (k_x + l_x + a)s} Y(s) \tag{14}$$

This can be interpreted as a standard PID controller in parallel form with a first-order low-pass filter on the derivative part:

$$G_c(s) = K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N}\right) \tag{15}$$

Again, for a given process parameter a, there is a direct correspondence between the LQG design parameters q_y, q_x, r_z, r_x and the PID parameters K, T_i, T_d, N. The formulas for the PID parameters, which are surprisingly unwieldy, are found in the Appendix.

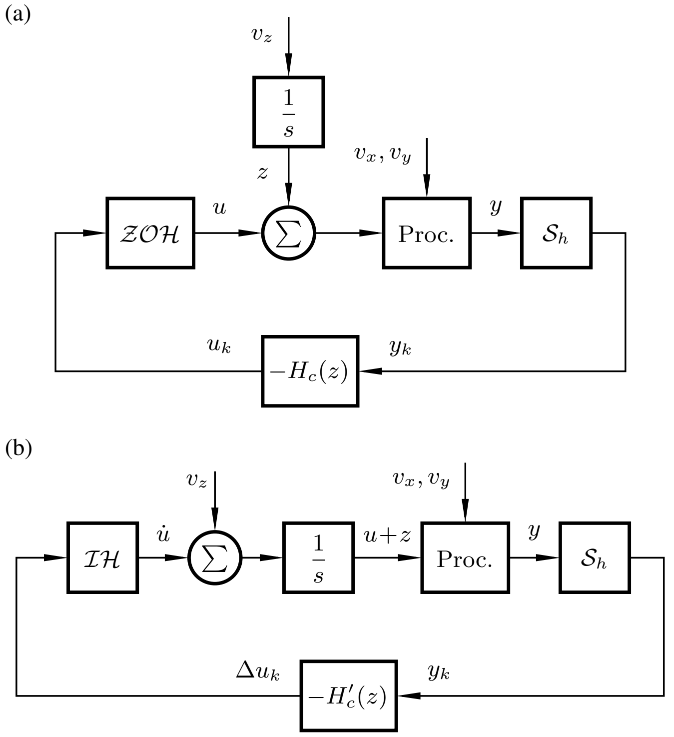

Fig. 4. Design in the sampled-data control case. (a) Standard sampled-data loop with periodic sampler $\mathcal{S}_h$, discrete-time controller $H_c(z)$, and zero-order hold $\mathcal{ZOH}$. (b) Equivalent sampled-data loop with incremental controller $H'_c(z)$, impulse hold $\mathcal{IH}$, and external integrator.

D. Optimal Sampled-Data Design

We now turn to the sampled-data case, see Fig. 4(a). We assume that the process output is sampled with a constant period h and that the controller output is held between updates using zero-order hold. The goal is to design a discrete-time controller $H_c(z)$ that minimizes the continuous-time cost function (1) in the PI case or (9) in the PID case.

The sampled-data LQG problem is nonstandard due to the unstable and uncontrollable input disturbance state. Similarly to [20], to simplify the design process we first perform a loop transformation into the equivalent formulation shown in Fig. 4(b). The input disturbance state and the controller integral state are merged into a single state, which is controllable. Making the integrator external to the controller, the goal is now to design an incremental controller $H'_c(z)$ with the discrete measurement y_k as input and the control signal movement Δu_k as output. The noise model and the process model, including the integral state, are then sampled assuming an impulse hold at the control signal input. After transformation and sampling, a standard discrete-time LQG design problem remains, which can be solved using standard control design software, yielding $H'_c(z)$. The final sampled-data controller, including the integrator, is then given by

$$H_c(z) = H'_c(z)\frac{z}{z-1} \tag{16}$$

The complete controller can be conveniently calculated using the `lqgdesign` command in the Jitterbug toolbox[1] [21].

[1] Jitterbug can be downloaded at http://www.control.lth.se/jitterbug

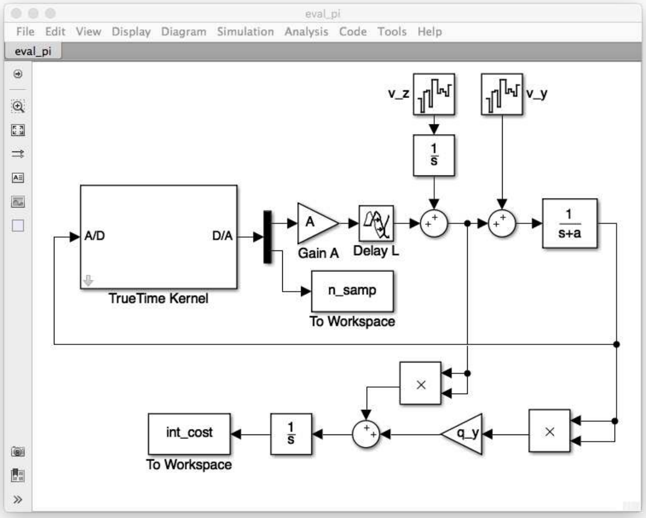

Fig. 5. Simulink model for Monte Carlo evaluation of the PI control benchmark.

E. Monte Carlo Evaluation of Event-Based Schemes

Unlike the continuous-time and sampled-data cases, the performance and event rates under event-based control schemes can typically not be computed analytically. As a general approach, we propose to evaluate the event-based PI and PID benchmarks using Monte Carlo simulations.

A Simulink model for Monte Carlo evaluation of the PI control benchmark is shown in Figure 5. The noise processes v_z and v_y are modeled using *Band-Limited White Noise* blocks. For a fair comparison, the seed values of the noise blocks are kept the same in the evaluation of the different schemes. The cost function is evaluated by integration over the simulation duration. Similarly, the event frequency is evaluated by recording the number of events in the simulation and then dividing by the simulation time.

A *Gain* block with value A (default = 1) and a *Transport Delay* block with value L (default = 0) are included in the control signal path to enable experimental evaluation of the robustness of the system. In a robustness evaluation, the value of either parameter is gradually increased until the closed-loop system becomes unstable, and then the corresponding amplitude and delay margins A_m and L_m are recorded.

The event-based controller itself can be modeled in a number of ways; we have used a *TrueTime Kernel*[2] block [22] to implement the various control schemes in Matlab code.

III. Benchmark Example 1: PI Control with Sensor Event Limitations

As a first example we study PI control with sensor event limitations, see Fig. 6. In particular, we focus on the trade-off between the regulator cost (1) and the average number of sensor events per time unit. We assume the following stable first-order process ($a = 1$):

$$G_p(s) = \frac{1}{s+1}$$

[2]TrueTime can be downloaded at http://www.control.lth.se/truetime

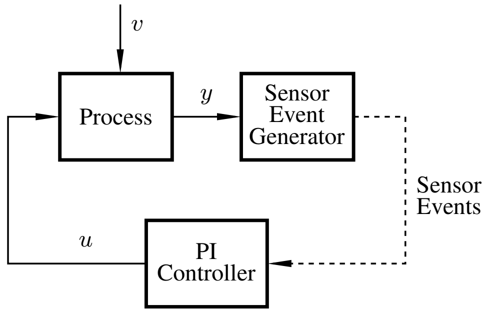

Fig. 6. Example 1: PI control with sensor event limitations.

Further, we assume the LQG design weights $q_y = r_z = 1$. In the subsections below, we shall evaluate and compare the following control schemes using the benchmark:

- Continuous-time PI control.
- Periodically sampled PI control.
- Send-on-delta sampling with Årzén's simple event-based PI controller [2].
- Send-on-delta sampling with Durand and Marchand's improved event-based PI controller [11]
- Send-on-delta sampling with state feedback from a particle filter.

A. Continuous-Time PI Control

For continuous-time control, the given process and design parameters yield the LQG-optimal gains $l_y = \sqrt{2} - 1$, $k_z = 1$ and the optimal PI controller parameters $K = \sqrt{2}$, $T_i = 1$. The bandwidth of the closed-loop system is $\sqrt{2}$ rad/s, and the optimal cost is $J^* = \sqrt{2}$. The performance of all controllers will be normalized to show the cost relative to this baseline. As for robustness, the amplitude margin is $A_m = \infty$ and the delay margin is $L_m = 1.11$ s.

B. Periodically Sampled PI Control

For the sampled-data PI controller, we explore the event–performance trade-off by adjusting the sampling period h between 0.05 and 2 s, corresponding to sampling rates between 0.5 and 20 Hz. For each value of h, the optimal sampled-data PI controller is designed and the cost function (1) is evaluated using Jitterbug [21]. Given a digital controller, it is trivial to evaluate the amplitude and delay margins of the sampled-data control system.

C. Årzén's Simple Event-Based PI(D) Controller

In Årzén's simple event-based PI controller, we set the nominal sampling interval to $h_{nom} = 0.05$ s. This is the interval at which the sensor event triggering condition is checked. The maximum sampling interval is set to $h_{max} = 2$ s. The control performance and the event rate are varied by setting the error limit e_{lim} to different values between 0.05 and 1.4. For each value of e_{lim}, the control performance and the average event rate are evaluated in a Monte Carlo simulation of length 2000 s.

The pseudo-code for the full PID controller is shown below. (For PI control, we simply set $T_d = 0$.)

```
%-- Arzen's simple event-based controller --

% Input and event detection logic
y = ADIn();
e = -y;
h_act = h_act + h_nom;
IF (abs(e - e_old) >= e_lim || h_act >= h_max) THEN
  % Calculate output
  u_p = K * e;
  a_d = T_d / (N * h_act + T_d);
  u_d = a_d * u_d - K * N * a_d * (y - y_old);
  u = u_p + u_i + u_d;
  % Output
  DAOut(u);
  % Update state
  u_i = u_i + K / T_i * h_act * e;
  e_old = e;
  y_old = y;
  h_act = 0;
END
```

D. Durand and Marchand's Event-Based PI(D) Controller

Durand and Marchand [11] suggested several improvements to Årzén's simple event-based controller, two of which we adopt here. First, their controller uses the backward difference in the I-part, which is more correct since h_{act} is known for the previous interval and not for the next sampling interval. Second, they propose an exponential gain reduction in the I-part for large sampling intervals, which removes the need for the safety limit h_{max}. The pseudo-code for the controller is:

```
%-- Durand and Marchand's event-based controller --

% Input and event detection logic
y = ADIn(1);
e = -y;
h_act = h_act + h_nom;
IF (abs(e - e_old) > e_lim) THEN
  % Calculate output
  u_p = K * e;
  a_d = T_d / (N * h_act + T_d);
  u_d = a_d * u_d - K * N * a_d * (y - y_old);
  h_act_i = h_act * exp(h_nom - h_act);
  u_i = u_i + K / T_i * h_act_i * e;
  u = u_p + u_i + u_d;
  % Output
  DAOut(1,u);
  % Update state
  e_old = e;
  y_old = y;
  h_act = 0;
END
```

The resulting regulator cost and average event frequency are evaluated in the same way as in the previous subsection.

E. State Feedback from Particle Filter

The final scheme that we evaluate for the first benchmark example is state feedback from a particle filter [23]. The particle filter attempts to emulate an optimal, Bayesian observer for the process with send-on-delta measurements. The particle filter is based on a discretized system model with the basic sampling interval $h = 0.05$. The non-Gaussian state probability distribution approximated by 2000 particles that evolve according to the stochastic process model. In each update step, each particle is assigned a weight proportional to its likelihood given the latest information (or lack of information) from the send-on-delta sensor. Based on the weighted mean of the particles, a point estimate $(\hat{y}, \hat{z})$ is calculated. The control signal is then calculated using the certainty equivalence feedback law

$$u = -l_y \hat{y} - l_z \hat{z} \tag{17}$$

The particles are regenerated in each step using the systematic resampling algorithm [23].

The resulting regulator cost and average event frequency are evaluated in the same way as in the previous subsection.

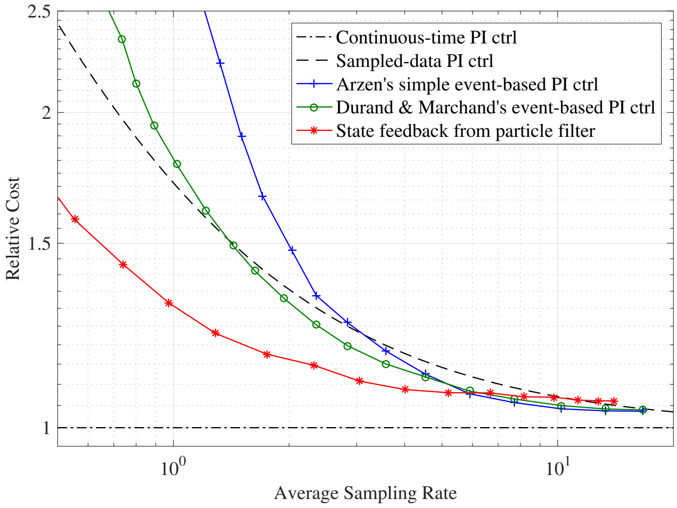

Fig. 7. Benchmark results for PI control with sensor event limitations.

F. Results

The performance results from the PI control benchmark are summarized in Fig. 7. Continuous PI control represents the baseline with a relative cost of 1. The sampled-data PI controller shows a gradual degradation in control performance as the sampling rate is decreased, as expected. Årzén's event-based PI controller beats the sampled-data controller at average rates above 3 Hz but performs worse than it at lower rates. Durand & Marchand's event-based PI controller shows some improvement and beats the sampled-data controller at average rates above 1.5 Hz. At very low average rates, state feedback from the particle filter clearly outperforms the other schemes. This is not surprising, given that the particle filter mimics an optimal event-based observer for the system. However, the amount of on-line computations are several orders of magnitude larger for this scheme compared to the other schemes. Also, the particle filter shows some performance degradation for high average rates. The overall performance of the particle filter can be improved by increasing the number of particles, but that would increase the computational demand even further.

Table I shows the robustness results for the different controllers. The sampled-data and event-based controllers have been tuned so that the nominal average sampling rate is 3 Hz. The amplitude margin A_m and the delay margin L_m for the event-based controllers were found experimentally by

TABLE I. EXPERIMENTAL ROBUSTNESS MEASURES FOR PI CONTROL WITH NOMINAL AVERAGE SAMPLING RATE 3 HZ.

	Cost	A_m	L_m
Continuous-time control	1.00	∞	1.11
Sampled-data control	1.24	5.3	1.24
Årzén's simple event-based PI control	1.22	29	1.15
Durand & Marchard's event-based PI control	1.15	28	1.17
State feedback from particle filter	1.11	32	2.10

gradually increasing the loop gain or the loop delay. It can be noted that the amplitude margins of the event-based controllers are larger than that of the sampled-data controller. The event-based controllers however generate more sensor events as the loop gain is increased, hence consuming much more resources. It can also be seen that state feedback from the particle filter is overall more robust than the other event-based schemes.

IV. BENCHMARK EXAMPLE 2: PID CONTROL WITH CONTROL EVENT LIMITATIONS

In the second example we study PID control with actuator event limitations, see Fig. 8. We now focus on the trade-off between the regulator cost (9) and the average number of control events per time unit. We assume that the process output can be sampled without cost and that the control signal generator is a zero-order hold circuit.

Setting $a = 1$, we have the following process to control:

$$G_p(s) = \frac{1}{s(s+1)}$$

We assume the LQG weights $q_y = r_z = 1$ and $q_x = r_x = 0$. In the subsections below, we shall evaluate and compare the following control schemes using the benchmark:

- Continuous-time PID control.
- Periodically sampled PID control.
- Send-on-delta sampling with Durand and Marchand's improved event-based PID controller [11].
- State feedback from Kalman filter with quantizer-based control event generator.
- State feedback from Kalman filter with send-on-delta control event generator.

A. Continuous-Time PID Control

For continuous-time control, the given process and design parameters yield the LQG-optimal gains $l_y = k_z = 1$ and $l_x = k_x = \sqrt{3} - 1$ and the optimal PID controller

$$G_c(s) = \frac{(6 - 2\sqrt{3})s^2 + 2\sqrt{3}\,s + 1}{s^2 + (2\sqrt{3} - 1)s}$$

The corresponding PID parameters are $K = 1.24$, $T_i = 3.06$, $T_d = 0.42$, $N = 1.04$. The bandwidth of the closed-loop system is 1.70 rad/s, and the optimal cost is $J^* = 5.32$. The performance of all controllers will be normalized to show the cost relative to this baseline. As for robustness, the amplitude margin is $A_m = \infty$ and the delay margin is $L_m = 0.81$ s.

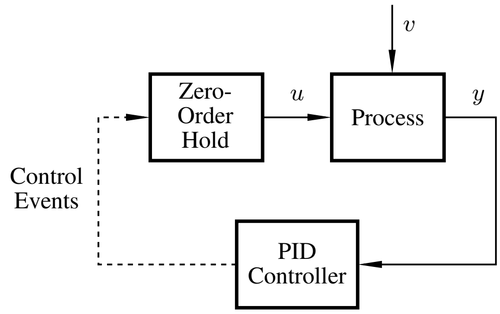

Fig. 8. Example 2: PID control with control event limitations.

B. Periodically Sampled PID Control

The sampled-data PID controller is designed and evaluated in a similar way as the sampled PI controller, see Section III-B.

C. Durand and Marchand's Event-Based PID Controller

For Durand and Marchand's improved event-based PID controller, we again use the nominal sampling interval $h_{nom} = 0.05$ s. Note that the controller always generates the same number of sensor and control events; the event-based PID algorithm is executed and the control signal is updated only when the input has changed by an amount of e_{lim}. By varying e_{lim} between 0.15 and 3 we explore the trade-off between performance and the control event rate. A Monte Carlo simulation of length 2000 s is performed for each value of e_{lim}.

D. Kalman Filter with Quantizer Control Event Generator

In this scheme, we assume that the process output y is sampled with the regular interval $h = 0.05$. (Recall that there is no cost on sensor events in this example.) The controller implements a standard discrete-time Kalman filter that produces the state estimates $\hat{z}$ and $\hat{x}$. In each sample, the controller evaluates the state feedback law (same as (10))

$$u = -l_y y - l_x \hat{x} - \hat{z} \tag{18}$$

The control signal is then sent through a quantizer with resolution Δ:

$$u_{quant} = \text{round}\left(\frac{u}{\Delta}\right) \cdot \Delta \tag{19}$$

A control event is generated and the zero-order hold control output is updated with the value u_{quant} only if the quantized control signal has changed value since the last sample. The actual zero-order hold control output is used in the Kalman filter to obtain correct state estimates. In the evaluation, Δ is varied between 0.15 and 3, and a Monte Carlo simulation of length 2000 s is performed for each value.

E. Kalman Filter with Send-on-Delta Control Event Generator

The final scheme that we evaluate is very similar to the previous one, except with regard to how control events are generated. Rather than quantizing the control output, we use a send-on-delta control generator. A new control signal is generated only if the value has changed by at least an amount Δ compared to the currently held control signal. When a

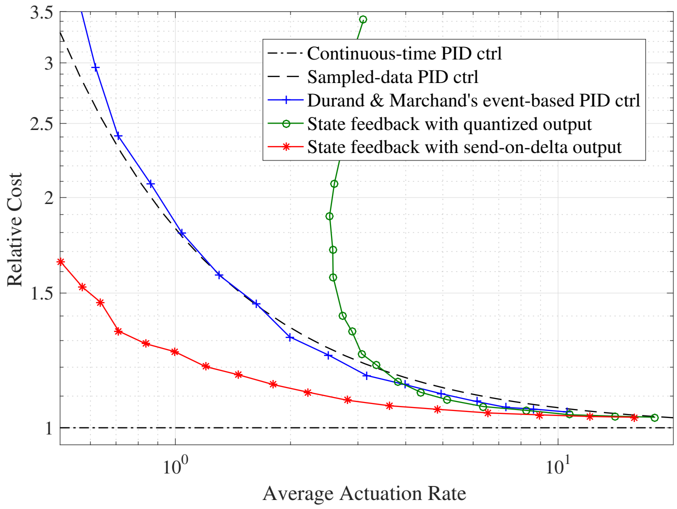

Fig. 9. Benchmark results for PID control with control event limitations.

control event is generated, the new value of u is sent to the zero-order hold, without quantization. As before, the actual control output is used in the Kalman filter to obtain correct state estimates. The controller is evaluated in the same way as in the previous subsection.

F. Results

The results from the PID control benchmark are shown in Fig. 9. The straight line with relative cost 1 represents the continuous-time PID controller. Like before, the sampled-data PID controller shows an expected performance degradation as the actuation rate is decreased. Durand and Marchand's event-based PID controller only shows a marginal improvement compared to the sampled-data controller. The quantized state feedback controller shows an improvement of similar magnitude for high actuation rates, but its performance breaks down for average rates below 3 Hz. The best event-based controller for this benchmark is the state feedback controller with the send-on-delta control generator. It retains good performance even for very low average actuation rates.

The experimental robustness results are reported in Table II. The sampled-data and event-based schemes have been tuned so that the nominal average control rate is 5 Hz. Again it is seen that the event-based schemes have good robustness towards unmodeled gains and delays. At the same time, they generate many more control events than the sampled-data controller as the stability limit is approached. The controller with the best performance—state feedback with send-on-delta output—also displays the best robustness properties.

TABLE II. EXPERIMENTAL ROBUSTNESS MEASURES FOR PID CONTROL WITH NOMINAL AVERAGE SAMPLING RATE 5 HZ.

	Cost	A_m	L_m
Continuous-time control	1.00	∞	0.81
Sampled-data control	1.13	9.3	0.80
Durand & Marchard's event-based PI control	1.11	32	0.77
State feedback with quantized output	1.09	18	0.79
State feedback with send-on-delta output	1.06	34	0.80

V. Conclusions and Future Work

We have presented two benchmarks problems for event-based control: One can be interpreted as a stochastic PI control problem, and the other one can be interpreted as a stochastic PID control problem. Even though we are dealing with low-order systems (two respectively three state variables in total), optimal solutions or even performance bounds are still unknown for many event-based problem formulations. The continuous-time case provides a lower bound that no other scheme can beat. The sampled-data case provides a bound that all event-based controllers should attempt to beat.

One thing to note is that the benchmarks assume constant-intensity white noise processes in the disturbance models. Hence, the processes are constantly being pushed away from their equilibria, and there are no periods of complete rest during which the event-based schemes can shine. Beating the sampled-data case may thus be non-trivial. The event-based PI(D) controllers investigated in the paper exhibit some problems at low event rates. There should hence be room for further algorithm improvements.

In the PI control benchmark, it was interesting to note that a nonlinear observer, such as a particle filter, can give a large performance increase under sensor event limitations. The particle filter is however very expensive in terms of computational effort, so this approach is probably not feasible for small embedded control systems.

In the PID control benchmark, it could be seen that a quantized control signal gave much worse performance than a send-on-delta output, which may seem surprising at first. With the quantizer, however, even a small random disturbance can cause the quantized output to change frequently, while the send-on-delta mechanism in itself implements a hysteresis function. Also, the send-on-delta output gives the correct output value at events, while the quantized output always contains a rounding error.

The experimental robustness evaluations showed that the event-based controllers display at least as good robustness properties as the sampled-data controllers.

For future work, it would be natural to evaluate and compare more event-based control schemes from the literature, in various settings. There are many variations of the general event-based control loop in Fig. 1, and each variation implies different design trade-offs and solution approaches. Further, it could be interesting to explore other noise models in the benchmark, such as shot noise, which may be more appropriate for event-based control benchmarks.

Acknowledgment

This work has been supported by the Swedish Research Council. The author is a member of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

References

[1] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *Preprints 14th World Congress of IFAC*, Beijing, P.R. China, July 1999.

[2] K.-E. Årzén, "A simple event-based PID controller," in *Preprints 14th World Congress of IFAC*, Beijing, P.R. China, July 1999.

[3] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, Sept 2007.

[4] M. Rabi, K. H. Johansson, and M. Johansson, “Optimal stopping for event-triggered sensing and actuation,” in *Proc. 47th IEEE Conference on Decision and Control*, Cancún, Mexico, Dec. 2008.

[5] R. Cogill, “Event-based control using quadratic approximate value functions,” in *IEEE Conference on Decision and Control*, 2009.

[6] J. Sijs and M. Lazar, “On event based state estimation,” in *Proc. Hybrid Systems: Computation and Control*, 2009, pp. 336–350.

[7] T. Henningsson, “Sporadic event-based control using path constraints and moments,” in *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, Florida, USA, Dec. 2011.

[8] G. M. Lipsa and N. C. Martins, “Remote state estimation with communication costs for first-order LTI systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2013–2025, Sept 2011.

[9] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, “Periodic event-triggered control for linear systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, April 2013.

[10] J. H. Sandee, “Event-driven control in theory and practice — trade-offs in software and control performance.” Ph.D. dissertation, Eindhoven Univ. of Techn., Netherlands, 2006.

[11] S. Durand and N. Marchand, “Further results on event-based PID controller,” in *Proc. European Control Conference*, 2009.

[12] V. Vasyutynskyy and K. Kabitzsch, “A comparative study of PID control algorithms adapted to send-on-delta sampling,” in *IEEE International Symposium on Industrial Electronics (ISIE)*, July 2010.

[13] U. Tiberi, J. Araujo, and K. H. Johansson, “On event-based PI control of first-order processes,” in *2nd IFAC Conference on Advances in PID Control*, 2012.

[14] A. Ruiz, J. E. Jiménez, J. Sánchez, and S. Dormido, “A practical tuning methodology for event-based PI control,” *Journal of Process Control*, vol. 24, no. 1, pp. 278–295, 2014.

[15] S. Durand and J. F. Guerrero-Castellanos, “Event-based digital PID control,” in *International Conference on Event-based Control, Communication, and Signal Processing*, June 2015.

[16] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *IEEE Conference on Decision and Control*, Dec 2012.

[17] Q. Liu, Z. Wang, X. He, and D. Zhou, “A survey of event-based strategies on control and estimation,” *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 90–97, 2014.

[18] A. Cervin and K. J. Åström, “On limit cycles in event-based control systems,” in *Proc. 46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 2007.

[19] K. J. Åström and T. Hägglund, *Advanced PID Control.* ISA – The Instrumentation, Systems and Automation Society, 2006.

[20] B. Lennartsson and R. Middleton, “Optimal anti-aliasing filter based on multi criteria sampled-data $\mathcal{H}_\infty$ control,” in *Proc. 53rd IEEE Conference on Decision and Control*, Los Angeles, CA, 2014.

[21] B. Lincoln and A. Cervin, “Jitterbug: A tool for analysis of real-time control performance,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[22] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, “How does control timing affect performance?” *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 16–30, June 2003.

[23] T. B. Schön, “Solving nonlinear state estimation problems using particle filters – an engineering perspective,” Division of Automatic Control, Linköping University, Sweden, Tech. Rep. LiTH-ISY-R-2953, May 2010.

APPENDIX

The LQG-optimal PID parameters in (15) are given by

$$
\begin{aligned}
K &= l_y - \frac{l_x l_y - k_z(a+l_x)}{a+k_x+l_x} - \frac{k_z l_y}{(a+k_x+l_x)^2} \\
T_i &= \frac{a^2+2ak_x+l_x a+k_x^2+l_x k_x-k_z}{k_z(a+k_x+l_x)} + \frac{a^2+2al_x+k_x a+l_x^2+k_x l_x}{l_y(a+k_x+l_x)} \\
T_d &= \frac{(l_y+ak_x+l_x k_x+k_x^2)(k_z+al_x+k_x l_x+l_x^2)}{(a+k_x+l_x)(a^2k_z-k_z l_y+a^2 l_y+k_x^2 l_y+k_z l_x^2+ak_x k_z+2ak_x l_y+2ak_z l_x+al_x l_y+k_x k_z l_x+k_x l_x l_y)} \\
N &= \frac{(l_y+ak_x+k_x l_x+k_x^2)(k_z+al_x+k_x l_x+l_x^2)}{a^2k_z-k_z l_y+a^2 l_y+k_x^2 l_y+k_z l_x^2+ak_x k_z+2ak_x l_y+2ak_z l_x+al_x l_y+k_x k_z l_x+k_x l_x l_y}
\end{aligned}
\tag{20}
$$