LQG-Optimal versus Simple Event-Based PID Controllers

Cervin, Anton; Thelander Andrén, Marcus

Published in:
2020 American Control Conference (ACC)

DOI:
10.23919/ACC45564.2020.9147559

2020

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

Link to publication

Citation for published version (APA):
Cervin, A., & Thelander Andrén, M. (2020). LQG-Optimal versus Simple Event-Based PID Controllers. In *2020 American Control Conference (ACC)* IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.23919/ACC45564.2020.9147559

Total number of authors:
2

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

LQG-Optimal versus Simple Event-Based PID Controllers

Anton Cervin and Marcus Thelander Andrén

***Abstract*— In this paper, we study event-based PID control from an optimal stochastic control perspective. The purpose is to better understand what implementation features are critical for achieving good event-based PID performance. For this end, we formulate an LQG control design problem for a double integrator process with an integral disturbance, where the solution is an ideal PID controller. We then consider the trade-off between LQG cost and average sampling rate and give an interpretation of the optimal sampled-data controller and event-based sampling policy in terms of PID control. Based on insights from the optimal solution, we finally discuss how suboptimal but simple event-based PID controllers can be implemented. The proposed implementation is evaluated in a simulation study and compared to earlier work in event-based PID control. The results highlight the importance of considering both the triggering rule and the transmitted information in order to obtain an event-based PID controller with good performance.**

I. Introduction

Event-based feedback control has a history reaching back to at least the middle of the last century, but the field has received renewed interest since the publication of [1] and [2], now two decades ago. The former paper studied optimal event-based impulse control of a first-order stochastic system, and analyzed the trade-off between average sampling rate and output variance. The latter proposed a simple event-based PID controller and evaluated its performance in simulations. Since then, several theoretical and practical research studies of event-based control have been performed, see the survey papers [3] and [4].

One possible structure for an event-based controller is shown in Fig. 1. Typically, we are interested in the trade-off between regulatory performance and the average event rate (corresponding to, e.g., network usage) in the loop.

A proposal to close the gap between theoretical, optimal event-based control and practical, heuristic event-based PID control via a simple benchmark problem was put forward in [5]. In this paper, we continue that development and study a modified benchmark problem from both LQG and PID perspectives. The work is prompted by recent theoretical breakthroughs by Mirkin and co-authors [6]–[8], who have provided an LQG-optimal controller structure under intermittent sampling. The resulting optimal controller may in general not be practical to design nor to implement, but it provides a lower bound on the achievable LQG cost with any event-based controller.

There are two main aspects to consider in the design of event-based control: (i) the rule for triggering events and (ii) control signal generation in between events. The design of (i) is the topic of several works, ranging from simple but well-known rules such as send-on-delta [9]–[11] and integral triggering [12], [13], to more complex model-dependent rules [7], [14]. While model-based rules often achieve better performance in theory, they are also more difficult to implement in practice. This is also true for (ii) where options range from a simple zero-order hold to more involved signal generators (a.k.a. generalized holds) [15], [16]. One of the main motivators of this paper is the study of reasonable choices of (i) and (ii) for event-based PID, striking a balance between performance and simplicity of implementation.

This work has been supported by the Swedish Research Council, grant no. 2017-04491. The authors are with the Department of Automatic Control, Lund University, Sweden, and are members of the ELLIIT Excellence Center. E-mail: {anton,marcus}@control.lth.se

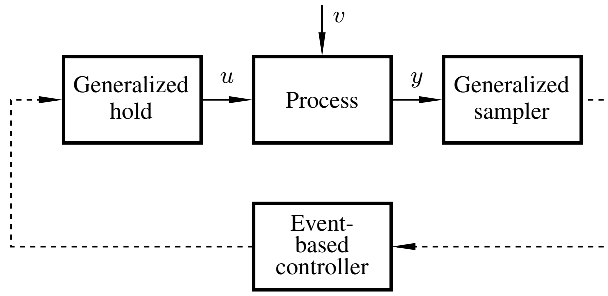

Fig. 1. An event-based controller structure for a single-input–single-output process. The solid lines represent continuous signal transmission, while the dashed lines indicate event-based communication.

The main contributions of this paper are

- the formulation of an LQG design problem, where the optimal solution is an ideal PID controller;
- an interpretation of the optimal sampled-data controller and event-based sampling policy in terms of PID control;
- a numerical evaluation of some common heuristic triggering rules and control generators in comparison to the optimal solution.

The rest of this paper is outlined as follows. In Sec. II, we formulate the LQG design problem that results in an ideal PID controller. In Sec. III we review Mirkin's LQG-optimal sampled-data controller structure and give an interpretation of it for the considered design problem in terms of PID control. In Sec. IV we discuss various heuristic event-based PID implementations and how they relate to the optimal solution. A performance comparison is presented in Sec. V, where the benefit of different controller structures and sampling policies are evaluated. Finally, Sec. VI concludes the paper and suggests some future work.

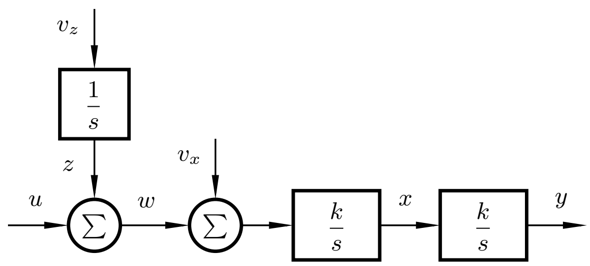

Fig. 2. Double integrator process with integral input disturbance.

II. The LQG Design Problem

As a benchmark problem, consider the system in Fig. 2, consisting of a double integrator and an input disturbance integrator. This is a simple control model, but relevant for, e.g., mechanical systems such as a satellite or a cart on a rail. We will first show how an LQG design problem for the system can be formulated such that the solution is an ideal PID controller. While a similar problem for a marginally stable process was proposed in [5], a drawback of that setup was the very complicated expressions for the resulting PID controller with a derivative filter. Further, not all PID controllers could be interpreted as LQG controllers, i.e., the inverse problem was not well-defined. Here we choose a different process and cost function, which produces a simpler solution and also permits inverse calculations (i.e., from given PID parameters to an LQG problem) in all cases. The derivation of the LQG controller below is trivial and we refer to, e.g., [17] for further details.

A. Translation between LQG and PID

For the system in Fig. 2, assume that $k > 0$ is a scalar gain parameter and that v_z and v_x are independent continuous-time white noise processes with intensities $r_z > 0$ and 1, respectively. The control objective is to minimize the cost function

$$J = \mathrm{E}\left\{ q_y y^2 + 2q_{yw} yw + w^2 \right\}, \tag{1}$$

where $q_y > 0$ and $q_{yw}^2 \leq q_y$ are scalar weights. Note that we penalize $w = z + u$ rather than the control signal u in order to allow the controller to have true integral action.

Assuming that the states z, x, and y are available for continuous feedback, the linear-quadratic control law (e.g., [17]) is given by

$$u = -l_x x - l_y y - z,$$

where the feedback gains l_x and l_y are given by the solution of the associated algebraic Riccati equation, yielding

$$l_x = \sqrt{2\big(\sqrt{q_y} - q_{yw}\big)}, \tag{2}$$

$$l_y = \sqrt{q_y}. \tag{3}$$

Next we consider optimal estimation of the state vector. Since there is no measurement noise on y, we immediately have $x = \dot{y}/k$. The lack of process noise on x allows us to formulate a reduced-order Kalman–Bucy filter for z as

$$\dot{\hat{z}} = k_z(\ddot{y}/k^2 - u - \hat{z}), \tag{4}$$

where the optimal Kalman gain k_z is obtained by solving the associated Riccati equation, yielding

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

The complete LQG controller is then given by

$$\begin{aligned} \dot{\hat{z}} &= k_z(\ddot{y}/k^2 - u - \hat{z}), \\ u &= -l_x \dot{y}/k - l_y y - \hat{z}. \end{aligned} \tag{6}$$

At first glance, it may seem that the controller needs access to $\ddot{y}$. However, in input–output form the controller can be written as

$$u = -\frac{(k_z + kl_x)p^2 + (k^2 l_y + kk_z l_x)p + k^2 k_z l_y}{k^2 p}\, y, \tag{7}$$

where $p := \frac{d}{dt}$ is the differential operator. Comparing this to an ideal PID controller in parallel form,

$$u = -K\Big(1 + \frac{1}{pT_i} + pT_d\Big)y,$$

we obtain the algebraic relationships

$$K = \frac{l_y k + k_z l_x}{k}, \tag{8}$$

$$T_i = \frac{l_y k + k_z l_x}{k_z l_y k}, \tag{9}$$

$$T_d = \frac{k_z + l_x k}{l_y k^2 + k_z l_x k}. \tag{10}$$

B. Interpretation of the LQG Controller

The LQG controller (6) is not suitable for implementation as it stands since it has $\ddot{y}$ as an input. A change of variables,

$$\hat{z}_i = \hat{z} - \frac{k_z l_x}{k} y - \frac{k_z}{k^2}\dot{y},$$

separates out the integrator state, $\hat{z}_i$, and moves the direct terms into the state feedback law. The resulting observer can be written as

$$\dot{\hat{z}}_i = -k_z\Big(u_{pi} + \frac{k_z l_x}{k} y + \hat{z}_i\Big), \tag{11}$$

where the control signal $u = u_{pi} + u_d$ has been split into a PI part and a D part with

$$\begin{aligned} u_{pi} &= -Ky - \hat{z}_i, \\ u_d &= -KT_d \dot{y}. \end{aligned} \tag{12}$$

In this formulation $\dot{y}$ does not enter the integrator, which is reasonable.

The observer (11) can be further modified by introducing a parameter $0 \leq \alpha \leq 1$ to split the PI control signal as

$$u_{pi} = \alpha u_{pi} + (1-\alpha)u_{pi} = -\alpha K y - \alpha \hat{z}_i + (1-\alpha)u_{pi},$$

yielding the family of possible observers

$$\dot{\hat{z}}_i = -k_z\Big((1-\alpha)(u_{pi} + \hat{z}_i) + \big(\tfrac{k_z l_x}{k} - \alpha K\big)y\Big).$$

With $\alpha = 0$ we retain (11), while with $\alpha = 1$ we recover the standard integrator formula that only uses y:

$$\dot{\hat{z}}_i = \frac{K}{T_i} y. \tag{13}$$

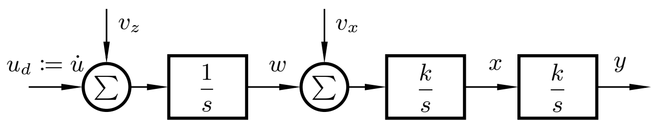

Fig. 3. Modified version of the system in Fig. 2, where the input disturbance and control signal have been merged into a single, controllable state.

With $\alpha = \frac{k_z l_x}{kK}$ we obtain the observer

$$\dot{\hat{z}}_i = -\frac{1}{T_i}\big(u_{pi} + \hat{z}_i\big), \tag{14}$$

which can be recognized as the classical "automatic reset" realization of integral action (see, e.g., [18]).

In a continuous feedback setting, all of the above observers behave identically, but in an event-based implementation they may yield different results depending on when and how the variables are communicated between the sensor, controller, and actuator.

To further guide the design of event-based implementations, we continue in the next section with a review of Mirkin's LQG-optimal sampled-data controller.

III. Mirkin's LQG-Optimal Controller

The LQG-optimal controller under any given sampling sequence was originally derived by Mirkin in [6], and subsequently adapted to event-based sampling in [7], [8]. The optimal solution retains elements of the continuous-time implementation in the form of an LQ control law and a Kalman-Bucy filter. We will here review the optimal design applied to the double integrator system in Fig. 2, where the controller elements can be interpreted in terms of PID control.

A. Preliminaries

The design of the optimal controller assumes a well-posed $\mathcal{H}_2$ design problem according to the conditions in [19, Sec. 14.5]. However, the system in Fig. 2 does not satisfy these conditions due to the uncontrollable state z and lack of measurement noise. We will therefore instead consider the design for a slightly modified problem, which in the limit will be equivalent to the original one.

The first modification is to transform the system in Fig. 2 into the system shown in Fig. 3. This is done by regarding the controller integrator as part of the process, and then merging the input disturbance state z and the control signal u into the controllable state $w = z + u$. The "new" input to the system is then the derivative of the original input, $u_d := \dot{u}$. Note that, so far the original and modified design problems are equivalent, since regardless of whether the system in Fig. 2 or 3 is considered, the optimal control design will still result in the same closed-loop system and cost.

Secondly, we proceed similarly to [7] and add an (artificial) white noise signal v_y with intensity $r_y > 0$ to the system output y and a small artificial penalty $q_{u_d} > 0$ to the input signal u_d. As long as r_y and q_{u_d} are close to zero, the modified design problem will only differ slightly, and in the limit $r_y, q_{u_d} \to 0^+$ the original problem is recovered.

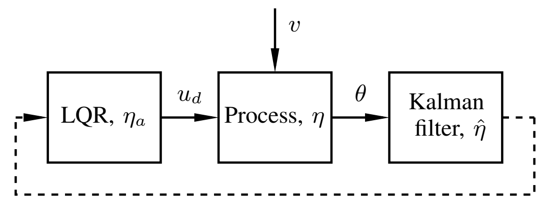

Fig. 4. Representation of the LQG-optimal sampled-data controller structure. The vectors η, η_a and $\hat{\eta}$ are the state vectors in each subsystem respectively. Solid and dashed lines represent continuous and sampled signals respectively.

With these modifications the design problem is well-posed and can be summarized by the following generalized process with state vector $\eta = [x,\, y,\, w]^\intercal$, noise vector $v = [v_x,\, v_y,\, v_z]^\intercal$, cost vector ξ and system output θ:

$$\begin{cases} \dot{\eta} &= A\eta + B_v v + B_{u_d} u_d, \\ \dot{\xi} &= C_\xi \eta + D_{\xi u_d} u_d, \\ \theta &= C_\theta \eta + D_{\theta v} v. \end{cases} \tag{15}$$

The system parameters in (15) are given by

$$A = \begin{bmatrix} 0 & 0 & k \\ k & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ B_v = \begin{bmatrix} k & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ B_{u_d} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$
$$C_\xi = \begin{bmatrix} 0 & q_{yw} & 1 \\ 0 & \sqrt{q_y - q_{yw}^2} & 0 \\ 0 & 0 & 0 \end{bmatrix},\ D_{\xi u_d} = \begin{bmatrix} 0 \\ 0 \\ \sqrt{q_{u_d}} \end{bmatrix},$$
$$C_\theta = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix},\ D_{\theta v} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

The cost in the modified design problem is given by

$$J_{\text{mod}} = \mathrm{E}\left\{\xi^\intercal \xi\right\} = \mathrm{E}\left\{q_y y^2 + 2q_{yw} yw + w^2 + q_{u_d} u_d^2\right\}. \tag{16}$$

B. The Optimal Controller

While the optimal solution has a realization divided into a discrete-time controller and a generalized sampler and hold [7, Remark 2], we will here instead opt for the more intuitive realization shown in Fig. 4. The controller is then divided into a Kalman-Bucy filter on the sensor side,

$$\dot{\hat{\eta}} = A\hat{\eta} + B_{u_d} u_d + K_s(\theta - C_\theta \hat{\eta}), \tag{17}$$

and an LQR signal generator on the actuator side. The control signal is based on a continuous-time simulation of the process, whose state vector η_a is reset to the current state-estimate $\hat{\eta}$ at each sampling time t_i:

$$\begin{cases} \dot{\eta}_a &= (A - B_{u_d} L_a)\eta_a, \qquad \eta_a(t_i) = \hat{\eta}(t_i), \\ u_d &= -L_a \eta_a. \end{cases} \tag{18}$$

The vectors K_s and L_a are the Kalman–Bucy filter and LQR gains respectively from the corresponding continuous-time LQG controller.

As $r_y, q_{u_d} \to 0^+$ we retain the original design problem. The simulated LQR together with the input integrator on the actuator side are then reduced to an ideal PID controller

$$u = -Ky_a - z_{i,a} - KT_d \dot{y}_a,$$

where the "measurement" y_a and integral action $z_{i,a}$ are generated by the intermittently reset simulation

$$\ddot{y}_a + kl_x\dot{y}_a + k^2 l_y y_a = 0, \quad y_a(t_i) = y(t_i),\ \dot{y}_a(t_i) = \dot{y}(t_i),$$
$$\dot{z}_{i,a} = \frac{K}{T_i} y_a,\ z_{i,a}(t_i) = \hat{z}_i(t_i). \tag{19}$$

On the sensor side, the signals y and $\dot{y}$ are directly available, while $\hat{z}_i$ is given by (13). At sampling times $\{t_i\}$, the data $(y,\, \dot{y},\, \hat{z}_i)$ is transmitted to the actuator side and resets the simulation according to (19).

Naturally, this continuous-time scheme is difficult to realize in practice, but from a theoretical point of view it provides a useful performance baseline for comparisons to more practical implementations.

C. Event-Based Sampling

Define the error $\tilde{\eta} := \eta_a - \hat{\eta}$. Whenever $\tilde{\eta} = 0$ holds, the optimal sampled-data controller will behave identically to its continuous-time counterpart. This is the case just after sampling, but in between sampling actions the error will drift due to disturbances in the system. The dynamics of $\tilde{\eta}$ will fundamentally determine the closed-loop performance, and it can be shown that (16) can be re-expressed as [8, Thm. 1]

$$J_{\text{mod}} = \gamma_0 + q_{u_d}\,\mathrm{E}\{\left(L_a\tilde{\eta}\right)^2\}, \tag{20}$$

where γ_0 is continuous-time LQG optimal cost. The value of the second term in (20) is determined by the choice of sampling policy, which ideally should be as small as possible for a given average sampling rate.

As detailed in [14], the event-based sampling policy that minimizes (20) is in the form of a threshold on $\tilde{\eta}$. Finding the optimal threshold for a given setup generally requires computationally demanding numerical methods. However, as seen in Fig. 5, the optimal threshold can in this case be well approximated by two parallel hyperplanes, orthogonal to the vector $[l_x\,, l_y,\, 1]^\intercal$. This approximation corresponds to a policy which triggers sampling whenever

$$|\tilde{u}| > \Delta. \tag{21}$$

where $\tilde{u} := K(y - y_a) + KT_d(\dot{y} - \dot{y}_a) + (\hat{z}_i - z_{i,a})$ is the difference in control signal between two ideal PID controllers with feedback from the true process and the simulation (19) respectively.

We will use (21) as the (near optimal) event-based threshold policy for the optimal controller structure, where the design parameter $\Delta > 0$ is chosen as a trade-off between LQG-cost and average sampling rate.

IV. Simple Event-Based PID Implementations

Event-based implementations of PID controllers are usually motivated by improved resource efficiency, especially in networked control systems, where savings in energy and bandwidth can be achieved by transmitting data less often. However, the computational capacity in the sensor and actuator nodes are usually limited in embedded implementations, which makes complex triggering conditions and

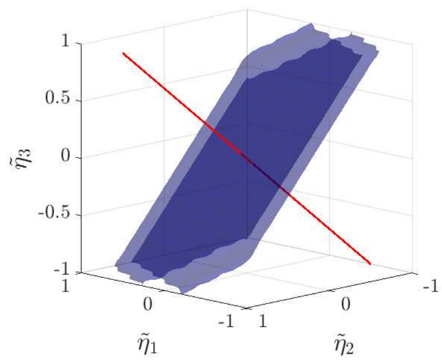

Fig. 5. Numerical approximation of the optimal sampling threshold (blue surface) for the LQG-optimal controller described in Sec. III. A line parallel to the LQ feedback gain vector $[l_x,\, l_y,\, 1]$ (red) is plotted for reference. The threshold was obtained using the method described in [20].

signal generators infeasible. Arguably, this is the case for the optimal controller of the previous section, which motivates the need for suboptimal but simpler implementations. In this section we will highlight some features of previously proposed event-based PID controllers from the literature, and discuss which features are useful yet practical to achieve good LQG performance.

A. Årzén's Simple Event-Based PID Controller

Most proposals of event-based PID controllers in the literature can be traced back to the seminal paper of Årzén [2]. We therefore start by giving a brief review of the algorithm here.

On the sensor side, the system output y is monitored periodically with a fixed, short period h_{nom}, and the decision to transmit data to the controller on the actuator side is based on a simple send-on-delta condition combined with a time-out h_{max}. The sensor operation is described by the following pseudo-code, in which `h_act` denotes the actual sampling period:

```
y := AnalogIn();
h_act := h_act + h_nom;
IF abs(y - y_old) >= delta OR h_act >= h_max THEN
   Send(y);
   y_old := y;
ENDIF
```

An on-the-fly discretized version of the PID algorithm is then implemented on the actuator side. It runs at each sensor event and is represented by the following pseudo code:

```
y := Receive();
h_act := Time() - time_old;
u_p := -K * y;
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;
AnalogOut(u);
u_i := u_i - K / T_i * h_act * y;
y_old := y;
time_old := Time();
```

Here, `N` represents the maximum derivative gain in the controller. If there is no measurement noise, we can let $\mathtt{N} \to \infty$ and the derivative part becomes a pure backward difference.

B. Choice of Measurement Filter and Triggering Rule

Årzén's event-based controller and many subsequent ones uses the send-on-delta rule [9], possibly after filtering out measurement noise [10]. It has been pointed out in several works, e.g., [12], [13] that integral sampling is less sensitive to noise and also eliminates the deadband effect.

The optimal solution contains a Kalman-Bucy filter at the sensor that filters out measurement noise and estimates the full state vector. It triggers on the difference in control signal between the sensor and actuator side, hence utilizing the entire state vector.

For PID control, it would seem like a reasonable compromise to trigger on the PD part of the control signal, i.e., whenever

$$\left|K(y - y_{\text{old}}) + KT_d(\dot{y} - \dot{y}_{\text{old}})\right| > \Delta.$$

This would also be relatively simple to implement in analog electronics, in conjunction with a second-order anti-aliasing filter (see [18]). We will refer to this option as *PD triggering*. To avoid stationary errors in the case of zero process noise and no events, the trigger needs to be combined with a time-out h_{max}, similar to Årzén's solution.

C. Choice of Data to Communicate

Most heuristic methods only communicate the sensor value y to the actuator side at events, although some works have proposed separate triggers and transmissions for the different parts of the PID controller, e.g., [11].

The optimal solution, however, transmits an estimate of the full state vector. Sending a few extra bytes in a network packet costs very little, and, if PD triggering is used, both y and $\dot{y}$ are already available in the sensor node and should be communicated to the actuator.

D. Choice of Integrator Implementation

Årzén's integrator implementation does not work well for long inter-event times. Durand & Marchand therefore proposed to include a forgetting factor to alleviate the problem [21]. Another solution is to implement the integrator in the form of an automatic reset, Eq. (14). This has for instance been adopted in the PIDplus commercial controller [22] and has been proven to work well for event-based PI control [13].

The optimal solution estimates the integral state as part of the full state vector. It is however of practical advantage to have the integrator separately, since it becomes easier to deal with practical issues such as anti-reset windup and controller mode switches [18].

E. Choice of Control Signal Generator

At the actuator side, zero-order hold (ZOH) between events is often assumed in heuristic implementations. Setting the correct feedback gain for each output however requires knowledge of the next hold interval. Better choices may therefore be impulse generators or general control signal generators as discussed in [16].

The optimal solution includes a full-state plant model at the actuator as a control generator. One compromise is to utilize a simplified model in the actuator [15]. A further possible solution is to use ZOH but adapt the feedback gain according to the recently experienced hold intervals. We will refer to this heuristic method as *adaptive PD gain*.

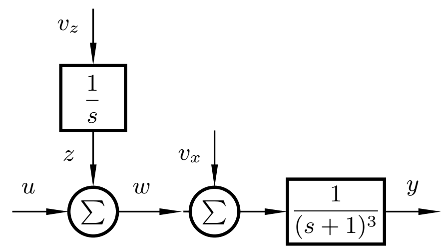

Fig. 6. Process considered for Setup B in the performance comparisons.

V. Performance Evaluation

In this section we evaluate the performance of several combinations of the heuristic event-based methods discussed in the previous section and compare the results to Mirkin's optimal controller with event-based sampling. Higher resource efficiency being one of the main motivators for event-based control, we focus on the trade-off between average sampling rate (equivalent to mean network or CPU usage) and the LQG cost as measured by (1). Two different setups are considered:

A. The double integrator process in Fig. 2 with the gain, cost, and noise parameters $k = 1$, $q_y = 4$, $q_{yw} = 0$, $r_z = 1$. The LQG-optimal controller is a PID controller with the parameters $K = 4$, $T_i = 2$, $T_d = 0.75$.

B. The stable third-order process in Fig. 6 with the cost and noise parameters $q_y = 5.5$, $q_{yw} = 0$, $r_z = 0.1$. For this higher-order process, the LQG-optimal controller is *not* a PID controller. We can however find the best possible PID parameters using nonlinear optimization, yielding $K = 2.15$, $T_i = 2.67$, $T_d = 1.23$.

Setup A matches the studied problem exactly, while Setup B is representative of a lag-dominated stable process from process industry. In both cases, the LQG costs found in the evaluation have been normalized so that the continuous PID controller has a relative cost of 1.0.

The performance of the different controllers were evaluated by Monte Carlo simulations in TrueTime [23]. Throughout, $h_{\text{nom}} = 0.01$ s was used as the simulation timestep (and hence smallest possible event detection interval). For each scenario, a 1000 s simulation was run using the same noise input sequences. For the event-triggered algorithms, the trigger parameter Δ was swept over a range of values, generating different average sampling rates and LQG costs.

A. Triggers and Sent Information

We first study how the choice of event trigger (send-on-delta *or* PD trigger) and the information sent between sensor and actuator (only y *or* y and $\dot{y}$) impact the performance of the event-based PID controller. The rest of the PID controller is implemented like Årzén's (see Sec. IV-A). Results for Case A are reported in Fig. 7. It is seen that PD triggering

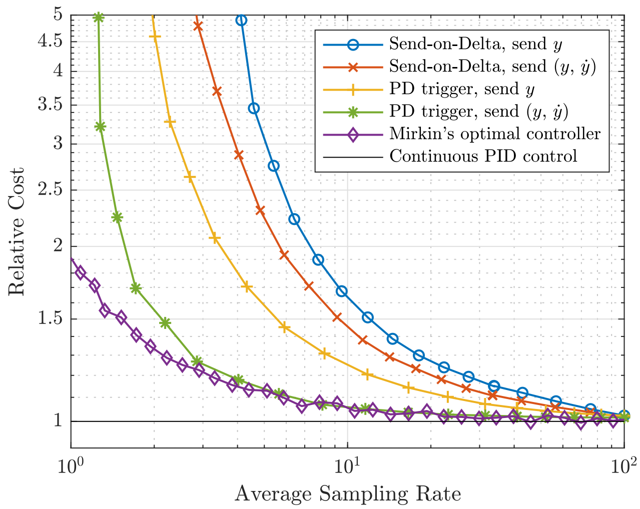

Fig. 7. Trade-off between LQG cost and average sampling rate for Setup A using different triggers and sent information.

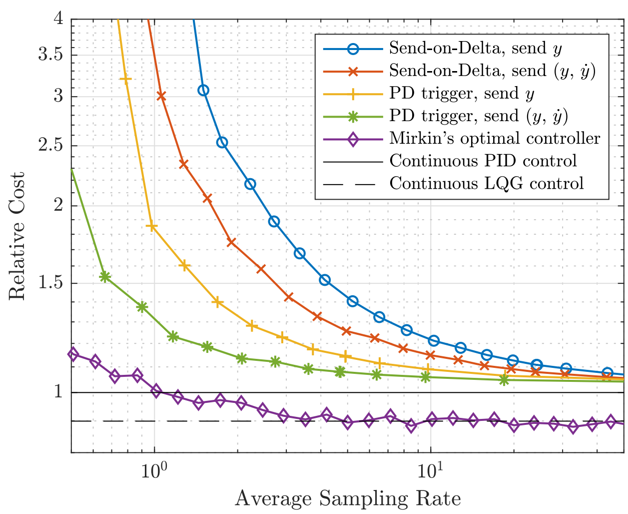

Fig. 8. Trade-off between LQG cost and average sampling rate for Setup B using different triggers and sent information.

gives a dramatic performance improvement over send-on-delta. The combination of PD triggering and sending both y and $\dot{y}$ gives close to optimal event-based performance for average event rates down to 1.2 Hz. The results for Setup B are quite similar as seen in Fig. 8. Note however the performance gap between the continuous-time PID and the LQG controllers in this case, as a PID controller cannot optimally control a third-order process under the given LQG criterion.

B. Integrator Algorithm and Feedback Gain

Keeping the controller with PD triggering and transmission of y from the previous subsection, we now try two alternative integral implementations: Durand & Marchand's exponential forgetting integral and the automatic reset implementation. We also experiment with fixed PD feedback gain versus an adaptive feedback gain that is adjusted based on the previous sampling interval. Results for the double integrator are reported in Fig. 9.

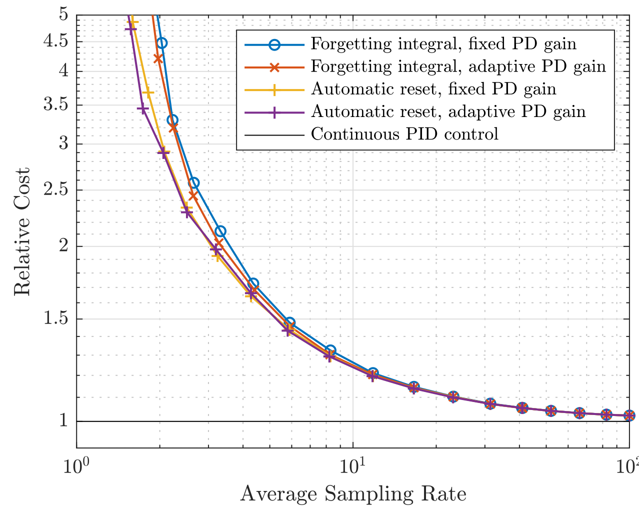

Fig. 9. Trade-off between LQG cost and average sampling rate for Setup A using different integrator algorithms and fixed or adaptive feedback gain.

Some small differences are visible, but overall the choice of integrator and PD gain implementation has quite a small impact on the performance. The automatic reset integrator performs slightly better than the forgetting integrator, while the adaptive gain seems to sometimes improve things and sometimes not. The results for Setup B are very similar and are omitted here.

C. Overall Comparison

In a final comparison, we study the range of possible performance under the various event-based PID controllers and the LQG-optimal periodic and event-based controllers. We also include the optimal periodic sampled-data controller [24] as a point of reference. The best heuristic event-based PID controller is achieved by a combination of PD event triggering, sending both y and $\dot{y}$ to the actuator, an automatic reset integrator, and a fixed PD gain. The results are shown in Figs. 10 and 11. For the double integrator, the performance of the best event-based PID controller is very close to Mirkin's optimal controller with event-based sampling. For the stable third-order system, a large performance improvement is also possible, but the distance to optimality is not exactly known since the optimal event-based controller in this case only provides a lower bound on the achievable cost. Overall it is seen that simple event-based controllers can be made to perform much better than periodic controllers if the proper implementation choices are made.

VI. Conclusion and Future Work

In this paper, we have built upon the work in [5] and studied a modified benchmark problem for which the LQG-optimal controller is a PID controller. Using the optimal sampled-data controller structure by Mirkin as a baseline, we have studied the LQG-optimal solution of the PID control problem to gain new insights on how a good but simple event-based PID controller could be implemented.

The results suggest that a practical yet well-performing event-based PID should trigger events not only on the

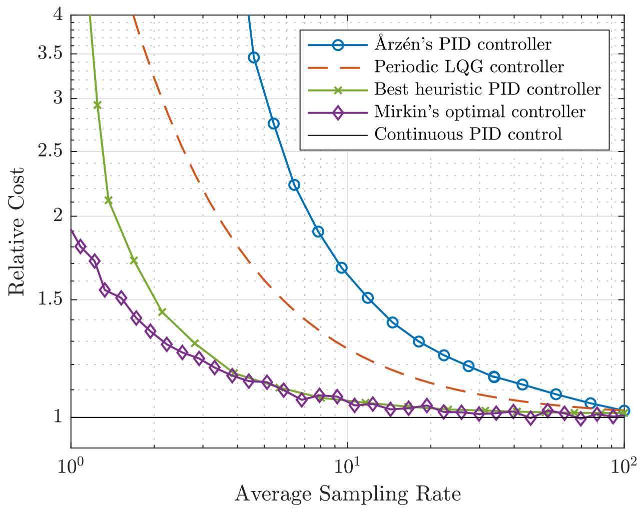

Fig. 10. Trade-off between LQG cost and average sampling rate for Setup A using different optimal and suboptimal PID and LQG controllers.

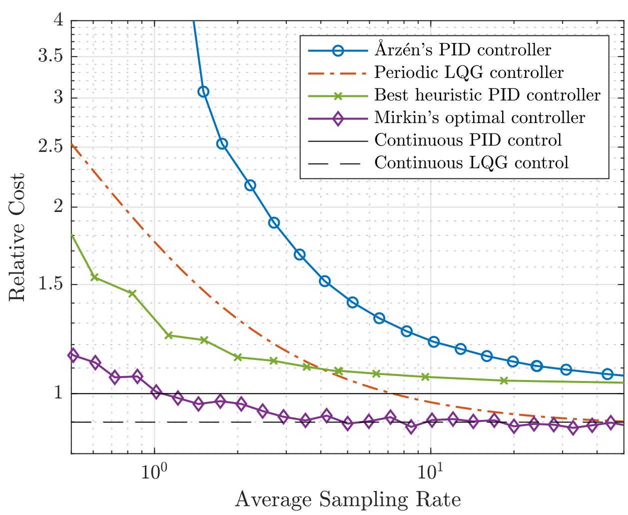

Fig. 11. Trade-off between LQG cost and average sampling rate for Setup B using different optimal and suboptimal PID and LQG controllers.

measurement y but also on its derivative $\dot{y}$, here referred to as "PD triggering". Furthermore, the sensor should also transmit both y and $\dot{y}$ to the controller at events. Small further improvements can be achieved if the integral action uses the automatic reset realization. Adapting the controller gain based on the length of the recent sampling period is probably not worth the extra implementation complexity, however.

One possible direction for further performance improvements is to experiment with other control signal generators than ZOH. The downside of more sophisticated signal generators is that they require a process model to be implemented, which is often not available in applications where PID control is considered. Another research direction would be to examine how well PD triggering actually approximates the optimal event trigger for the full-model system.

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, Jul. 1999.

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

[3] 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.

[4] 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.

[5] A. Cervin, "LQG-optimal PI and PID control as benchmarks for event-based control," in *Proc. 2nd International Conference on Event-Based Control, Communication and Signal Processing*, Krakow, Poland, 2016.

[6] L. Mirkin, "Intermittent redesign of analog controllers via the Youla parameter," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1838–1851, 2017.

[7] M. Braksmayer and L. Mirkin, "H2 optimization under intermittent sampling and its application to event-triggered control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7869–7874, 2017, 20th IFAC World Congress.

[8] A. Goldenshluger and L. Mirkin, "On minimum-variance event-triggered control," *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 32–37, 2017.

[9] M. Miskowicz, "Send-on-delta concept: An event-based data reporting strategy," *Sensors*, vol. 6, no. 1, p. 49, 2006.

[10] V. Vasyutynskyy and K. Kabitzsch, "Implementation of PID controller with send-on-delta sampling," in *International Conference Control*, 2006, pp. 4423–4428.

[11] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 4, 2002, pp. 3015–3020 vol.4.

[12] M. Miskowicz, "Asymptotic effectiveness of the event-based sampling according to the integral criterion," *Sensors*, vol. 7, no. 1, pp. 16–37, 2007.

[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] M. T. Andrén, B. Bernhardsson, A. Cervin, and K. Soltesz, "On event-based sampling for LQG-optimal control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 5438–5444.

[15] V. Vasyutynskyy and K. Kabitzsch, "First order observers in event-based PID controls," in *2009 IEEE Conference on Emerging Technologies Factory Automation*, Sep. 2009, pp. 1–8.

[16] K. Åström, "Event based control," in *Analysis and Design of Nonlinear Control Systems*, A. Astolfi and L. Marconi, Eds. Springer, 2008, pp. 127–147.

[17] K. J. Åström, *Introduction to stochastic control theory*, ser. Mathematics in science and engineering. Academic Press, 1970, vol. 70.

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

[19] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[20] M. T. Andrén, "Using radial basis functions to approximate the LQG-optimal event-based sampling policy," in *2019 18th European Control Conference (ECC)*, June 2019, pp. 2832–2838.

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

[22] J. Song, A. K. Mok, D. Chen, M. Nixon, T. Blevins, and W. Wojsznis, "Improving PID control with unreliable communications," in *ISA EXPO Technical Conference*, 2006, pp. 17–19.

[23] D. Henriksson, A. Cervin, and K.-E. Årzén, "TrueTime: Simulation of control loops under shared computer resources," in *Proceedings of the 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, Jul. 2002.

[24] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems (3rd Ed.)*. Upper Saddle River, NJ: Prentice-Hall, 1997.