From event-based PI and LQR strategies to
Event-based LQR with Integral Action

Sylvain Durand*

N. Marchand, J. Minet, J. Dumon, B. Boisseau, J.J. Martinez-Molina
* GIPSA-Lab, CNRS, Univ. Grenoble Alpes, Grenoble, France.

Faculty of Electronics, Autonomous University of Puebla (BUAP), Puebla, Mexico.

GdR MACS SDH

Paris, France
July 10, 2014
Research problematic and proposed solutions

Uniform vs. non-uniform sampling
Opportunity for embedded and networked control systems

Uniform sampling

1. Classical (time-triggered) control:
   → The control law is periodically computed and updated.
1. **Classical (time-triggered) control:**
   - The control law is periodically computed and updated.

2. **Event-triggered control:**
   - The control updates are event-driven (when a triggering condition is satisfied)
   - The sampling intervals are not equidistant in time anymore
   - The control signal is piecewise constant between events

---

**Uniform sampling**

- Setpoint
- Measured signal
- Sampling instants

**Non-uniform sampling**

- Setpoint
- Measured signal
- Sampling instants

$h_{nom}$

$q_{nom}$
Advantages of the non-uniform sampling scheme:

→ The number of samples can be highly reduced
   (and consequently the CPU utilization/communications)
   for the same final performance

⇒ This is clearly an opportunity for embedded and networked control systems
   (where resources are limited)
Event-Based LQR with Integral Action

1. Preliminary results on event-based PI control
2. More formal results on event-based LQR
3. Further results on event-based LQR with integral action
1 Preliminary results on event-based PI control
   - Time-triggered PI control
   - Event-based PI control
   - Experimental results: cruise control mechanism

2 More formal results on event-based LQR

3 Further results on event-based LQR with integral action
Classical PI control

\[(\text{Continuous-time) PI controller)}\]

\[u(t) = u_p(t) + u_i(t)\]  \hspace{1cm} (1)

\[\begin{align*}
  u_p(t) &= K_p e(t) \\
  u_i(t) &= K_i \int_0^t e(t) dt
\end{align*}\]

▷ \(u(t)\) is the control signal
▷ \(e(t) := y_{sp}(t) - y(t)\) is the error
▷ \(y_{sp}(t)\) is the setpoint to track
▷ \(y(t)\) is the measurement of the controlled system
▷ \(K_p\) and \(K_i\) are tunable parameters
The proportional part is straightforward

Backward difference approximation is applied for the integral part

\[
\begin{align*}
U(z) &= U_p(z) + U_i(z) \\
\text{with} \quad U_p(z) &= K_p E(z) \\
U_i(z) &= \frac{K_i \bar{h}}{1 - z^{-1}} E(z)
\end{align*}
\]

▷ $\bar{h}$ is the (constant) sampling period
Event-based PI control

The event-based PI architecture\[^2\] is divided into two parts:

1. A time-triggered event detector is used for level crossings
2. An event-triggered PID controller calculates the control signal

\[ h(t_i) \]

→ The event logic is time-triggered with the sampling period \( \bar{h} \)[\(^1\)]
   (that is the same as for the corresponding conventional time-triggered PI)

→ A request is sent when the absolute error crosses the detection level \( \bar{e} \)[\(^2\)]

→ The control signal is constant between two events

\[ u(t) = u(t_i) \quad \forall t \in [t_i, t_{i+1}], \quad i \in \mathbb{N} \]

- \( t_i \) is a sampling instant (called an \textit{event})
- Sampling intervals \( h(t_i) := t_i - t_{i-1} \) are not equidistant in time anymore

\[^1\] Årzén, "A simple event-based PID controller", IFAC WC 1999
\[^2\] Durand & al., "Further Results on Event-Based PID Controller", ECC 2009
What happens when the sampling interval varies?

i) The Nyquist-Shannon condition is no more consistent
ii) The approximations for discretization are no more accurate

⇒ Modifications are mandatory

- In particular for the integral gain

\[ u_i(t_i) = u_i(t_{i-1}) + K_i \underbrace{h(t_i)e(t_i)}_{\text{integral gain}} \]

...because it can lead to important overshoots when the control is updated after a large sampling interval

(somehow similar to the anti-windup mechanism used in control theory)
A solution is to **bound the integral gain** \([2,3]\)

\[
y_{sp} - y = e
\]

The product between the sampling interval \(h\) and the error \(e\) seriously increases after a steady state

...but, in fact, such an interval can be divided into two parts

**Bound of the integral part**

\[
u_i(t_i) = u_i(t_{i-1}) + K_i h_e(t_i)
\]

with \(h_e(t_i) \leq [h(t_i) - \bar{h}] \bar{e} + \bar{h} e(t_i)\)

---

1 A solution is to **bound the integral gain** \([2,3]\)

\[ e = y_{sp} - y \]

1 During the steady state, the sampling interval \( h \) increases but \( e < \bar{e} \)

2 When the setpoint changes, \( e \) becomes high but only during \( \bar{h} \)

**Bound of the integral part**

\[
u_i(t_i) = u_i(t_{i-1}) + K_i h e(t_i)
\]

with \( h e(t_i) \leq [h(t_i) - \bar{h}] \bar{e} + \bar{h} e(t_i)\)
Another solution is to apply an exponential forgetting factor of the sampling interval \[ [2] \]

\[
U_i(z) = \frac{K_i \lambda(h)}{1 - z^{-1}} E(z) \quad (3)
\]

with \( \lambda(h) = he^{\alpha(\bar{h} - h)} \)

\( \alpha \) is a degree of freedom

Other solutions are proposed in \([2]\)
Event-based (PI) cruise control mechanism [4]

Approximated model

\[ H(s) = \frac{G}{1 + \tau s} \]

AsynCar cruise control — Event-based PID control without safety limit condition (hybrid algorithm)

\[ \rightarrow \text{A reduction of about } 97\% \text{ of samples is achieved} \]
\[ \text{(in comparison with a classical time-triggered strategy)} \]

1 Preliminary results on event-based PI control

2 More formal results on event-based LQR
   - Formalization of the event-based paradigm
   - Event-based state-feedback (LQR) stabilization
   - Experimental results: attitude of a quadrirotor

3 Further results on event-based LQR with integral action
Let consider the linear time-invariant dynamical system

\[ \dot{x}(t) = Ax(t) + Bu(t) \quad (4) \]
\[ y(t) = Cx(t) \quad (5) \]

with \( x(0) := x_0 \)

- \( x \in \mathbb{R}^p \) is the state vector
- \( u \in \mathbb{R}^q \) is the control input vector
- \( y \in \mathbb{R}^r \) is the output vector
Definition (Event-based state-feedback)

By event-based state-feedback, we mean a set of two functions

1. an event function $\epsilon : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$
2. a state-feedback function $\upsilon : \mathbb{R}^p \to \mathbb{R}^q$ in the form $\upsilon(t) = -Kx(t)$

→ the control is computed and updated when $\epsilon \leq 0$
→ the control is kept constant when $\epsilon > 0$

$K$ is calculated to make the closed-loop system stable

$\Rightarrow$ The closed-loop solution of system (4) with event-based state-feedback $(\epsilon, \upsilon)$ starting in $x_0$ at $t = 0$ is then defined as the solution of the differential system

$$\dot{x}(t) = Ax(t) - BKx(t_i) \quad \forall t \in [t_i, t_{i+1}[ \quad i \in \mathbb{N}$$

$\triangleright$ $t_i$ is the last sampling time (the last time an event occurred)
$\triangleright$ $t_{i+1}$ is the next sampling time

Also, in the sequel let’s denote
$\triangleright$ $x_i := x(t_i)$ the state value at the last sampling time
Definition (Event-based state-feedback)

By event-based state-feedback, we mean a set of two functions

1. an event function \( \epsilon : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R} \)

2. a state-feedback function \( \upsilon : \mathbb{R}^p \to \mathbb{R}^q \) in the form \( \upsilon(t) = -Kx(t) \)

→ the control is computed and updated when \( \epsilon \leq 0 \)
→ the control is kept constant when \( \epsilon > 0 \)

- \( K \) is calculated to make the closed-loop system stable

\( \Rightarrow \) The closed-loop solution of system (4) with event-based state-feedback \((\epsilon, \upsilon)\) starting in \( x_0 \) at \( t = 0 \) is then defined as the solution of the differential system

\[
\dot{x}(t) = Ax(t) - BKx(t_i) \quad \forall t \in [t_i, t_{i+1}] \quad i \in \mathbb{N}
\]

- \( t_i \) is the last sampling time (the last time an event occurred)
- \( t_{i+1} \) is the next sampling time

Also, in the sequel lets denote
- \( x_i := x(t_i) \) the state value at the last sampling time
Property (Minimal inter-Sampling Interval)

1. An event-based control \((\epsilon, \upsilon)\) fulfils the Minimal inter-Sampling Interval (MSI) property if and only if there is some non zero minimal sampling interval.

2. An event-based control \((\epsilon, \upsilon)\) is said uniformly MSI if and only if there is some non zero minimal sampling interval for any initial condition \(x_0\).

\[ \rightarrow \] A MSI event-based control is a piecewise constant control with non zero sampling interval.

\[ \rightarrow \] A MSI event-based control is useful for implementation.
Consider the linear time-invariant system (4)

Take a (infinite horizon) quadratic cost functional defined by

\[ J = \int_{0}^{\infty} \left( x^T Q x + u^T R u \right) dt \] (6)

where \( P, Q, R \) are some positive definite matrix solution of the Riccati equation

\[ Q_1 - 2Q_2 = -Q \] (7)

with \( Q_1 := PA + A^T P \) and \( Q_2 := 2PBR^{-1}B^T P \)

Then the Lyapunov function

\[ V(x) := x^T Px \] (8)

is a CLF for the system (4)

since \( \forall x \neq 0, u = -2R^{-1}B^T Px \) renders \( \dot{V} \) strictly negative

(and, moreover, minimizes the value of (6))
Theorem (Event-based state-feedback stabilization)

Taking the CLF \( V \) in (8) for system (4), where \( P \) is a positive definite matrix solution of the Riccati equation (7), then the event-based state-feedback \((\epsilon, \nu)\) defined by

\[
\nu(t) = -Kx(t_i) \quad \forall t \in [t_i, t_{i+1}] \quad i \in \mathbb{N}
\]

with \( K := 2R^{-1}B^TP \)

\[
\epsilon(x, x_i) = (\sigma - 1)x^TQ_1x - 2x^TQ_2(\sigma x - x_i)
\]

with \( \sigma \in ]0, 1[ \)

is uniformly MSI and asymptotically stable\([5]\).

\( \rightarrow \) the higher \( \sigma \), faster is the convergence but more frequent are events\([5]\)

\( \rightarrow \) the smaller \( R \), larger is the control signal and smaller is the output of the controlled system\([6]\)

\[\text{Marchand \\ & al., "A General Formula for Event-Based Stab. of Nonlinear Systems", TAC'13}\]

\[\text{Téllez \\ & al., "Event-Based LQR Control for Attitude Stabilization of a Quadrotor", CLCA'12}\]
How is obtained the event function?

- An event is enforced when the stability is not ensured anymore, i.e. the derivative of the Lyapunov function \( \frac{dV}{dt} \) increases.

- In particular, the event function (10) can be rewritten as

\[
\epsilon(x, x_i) = -x^T Q_1 x - 2x^T Q_2 x_i + \sigma x^T (Q_1 - 2Q_2) x
\]

where

- \( i \) is the value of \( \frac{dV}{dt} \) when applying a constant control \( \nu(x_i) \) since the last event
- \( ii \) is the value of \( \frac{dV}{dt} \) if a continuously varying control \( \nu(x) \) was applied

→ This formula comes from the Sontag’s universal formula\(^7\)

---

\(^7\) Sontag, "A universal construction of Artstein's theorem on nonlinear stabilization", SCL 1989
Event-based (LQR) attitude stabilization \[6\]

Kinematic and dynamic equations

\[
\begin{align*}
(\dot{\phi} \; \dot{\theta} \; \dot{\psi})^T &= M\omega \\
J\dot{\omega} &= -\omega \times J\omega + \Gamma + G_a
\end{align*}
\]

\[
x = (\psi \; \theta \; \phi \; \dot{\psi} \; \dot{\theta} \; \dot{\phi})^T
\]

→ A strong reduction of the updates even when the system has to be actively actuated to be stabilized

---

1. Preliminary results on event-based PI control
2. More formal results on event-based LQR
3. Further results on event-based LQR with integral action
   - Event-based output-feedback stabilization
   - Trajectory tracking and integral action
   - Experimental results: gyroscope
The full state information $x$ is not measurable in practice
- Only a small number of outputs $y$ is really available
- A state observer is applied to estimate all the state information (possible as soon as $(A, C)$ is an observable pair)

Let consider the Luenberger state observer

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - C\hat{x}(t)] \quad (11)$$

with $\hat{x}(0) := \hat{x}_0$

- $\hat{x} \in \mathbb{R}^n$ is the estimated state vector
- $L$ is calculated to make stable the observation error

$$\tilde{x}(t) := x(t) - \hat{x}(t) \quad (12)$$
Definition (Event-based output-feedback)

By event-based output-feedback, we mean a set of two functions

1. an event function \( \epsilon : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R} \)
2. an output-feedback function \( \nu : \mathbb{R}^p \rightarrow \mathbb{R}^q \) in the form \( \nu(t) = -K\hat{x}(t) \)

\[ \Rightarrow \] The control is computed and updated when \( \epsilon \leq 0 \)

\[ \Rightarrow \] The control is kept constant when \( \epsilon > 0 \)

\[ \Rightarrow \] The closed-loop solution of system (4) with event-based output-feedback \((\epsilon, \nu)\) starting in \( x_0 \) at \( t = 0 \) is then defined as the solution of the differential system

\[
\dot{\hat{x}}(t) = A\hat{x}(t) - BK\hat{x}(t_i) + L[y(t) - C\hat{x}(t)] \\
\dot{x}(t) = Ax(t) - BK\hat{x}(t_i) \quad \forall t \in [t_i, t_{i+1}] \quad i \in \mathbb{N}
\]

\[ \Rightarrow \] The extension from state- to output-feedback consists in applying \( \hat{x} \) instead of \( x \) in the control law (9) and the event function (10)
Notation for trajectory tracking

\[ x_e(t) := x(t) - x_r(t) \]  \hspace{1cm} (14)
\[ e(t) := r(t) - y(t) \]  \hspace{1cm} (15)

- \( x_r \) is the state vector
- \( r \) is the output setpoint vector
- \( x_e \) is the state error vector
- \( e \) is the output tracking error vector

→ Rewriting system (4)-(5) gives

\[ \dot{x}_e(t) = Ax_e(t) + Bu(t) \]  \hspace{1cm} (16)
\[ y(t) = Cx_e(t) \]  \hspace{1cm} (17)

with \( x_e(0) := x_{e0} \)
Integral action

$\rightarrow$ Build extra states $z$ defined as the error integral

\[ \dot{z}(t) = e(t) \]  
with $z(0) := 0$  \hspace{2cm} (18)

$\rightarrow$ Construct the augmented system

\[ \dot{x}(t) = \bar{A}x(t) + \bar{B}u(t) + B_r r(t) \]  
\[ y(t) = \bar{C}x(t) \]  
with $x(0) := [x_e^T \ 0]^T$  \hspace{2cm} (19)-(20)

$\rightarrow$ Replace $x_e$ in the event-based feedback (9)-(10)
Exponential forgetting factor of the sampling interval \[2\]

\[ z(t_i) = z(t_{i-1}) + \lambda(h)e(t_i) \quad (21) \]

with \( \lambda(h) = he^{\alpha(\bar{h} - h)} \)

- \( \bar{h} \) is the (constant) sampling period
- \( \alpha \) is a degree of freedom

---

Gyroscope M750p [8]

Dynamic model

\[ \dot{\omega}_2 = \frac{J_D \Omega}{I_C + I_D} \omega_4 + \frac{1}{I_C + I_D} T_2 \]
\[ \dot{\omega}_3 = -\frac{1}{J_B + J_C} T_1 \]
\[ \dot{\omega}_4 = -\frac{J_D \Omega}{I_D + K_A + K_B + K_C} \omega_2 \]

\[ \Rightarrow \omega_1 = \Omega = 42 \text{ rad/s} \]

Actuators limits

\[ |T_1| \leq 0.2 \text{ Nm} \]
\[ |T_2| \leq 3.0 \text{ Nm} \]

Angles limits

\[ |\theta_i| \leq 20^\circ \quad \text{for } i = 2 \text{ to } 4 \]

\[ x = (\theta_3 \ \theta_4 \ \omega_2 \ \omega_3 \ \omega_4)^T \]
\[ u = (T_1 \ T_2)^T \]
\[ y = (\theta_3 \ \theta_4)^T \]

→ System in limit of stability, controllable and observable

[8] Educational control products (http://www.ecpsystems.com/)
Gyroscope M750p [8]

→ A tracking to non-zero setpoints
→ A strong reduction of the updates
→ A respect of the actuators limits

[8] Educational control products (http://www.ecpsystems.com/)
We developed:

1. an event-based PI controller \[^2\]
2. an event-based LQR \[^6\]
3. an event-based LQR with exp. forgetting factor of the integral action

We implemented on real-time systems:

1. Small radio-controlled car
2. Quadrirotor (attitude and position control)
3. Inverted pendulum
4. Gyroscope

\[^2\] Durand & al., "Further Results on Event-Based PID Controller", ECC 2009
\[^6\] Téllez & al., "Event-Based LQR Control for Attitude Stabilization of a Quadrotor", CLCA'12
We are now investigating:

1. the extension to nonlinear systems in the spirit of [5]
2. the extension to time-delay systems in the spirit of [9]

Thank you!