Lecture 12: Distributed and saturated event-based control

Lecture 12 Outline

• Distributed event-based control
• Anti-windup for event-based control
• Event-based PID control
Distributed event-based control

- How to implement event-based control over a distributed system?
  - E.g., control of multi-robot systems
- Local control and communication, but global objective

**Approach:** Consider a prototype distributed control problem and study it under event-based communication and control

---

Average consensus problem

**Multi-agent system model**
- Group of $N$ agents
  \[ \dot{x}_i(t) = u_i(t) \]
- Communication graph $G$
  \[ A: \text{undirected, connected} \]

**Adjacency matrix** $A$ with $a_{ii} = 1$ if agents $i$ and $j$ adjacent, otherwise $a_{ij} = 0$

**Degree matrix** $D$ is the diagonal matrix with elements equal to the cardinality of the neighbor sets $N_i$

**Objective:** Average consensus
\[ x_i(t) \xrightarrow{t \to \infty} a = \frac{1}{N} \sum_{i=1}^{N} x_i(0) \]

**Consensus protocol**
\[ u_i(t) = -\sum_{j \in N_i} (x_i(t) - x_j(t)) \]

**Closed-loop dynamics**
\[ \dot{x}(t) = -Lx(t) \]
with Laplacian matrix $L = D - A$

**Event-based implementation?**
### Event-based average consensus

**Event-based scheduling of measurement broadcasts:**

- Consensus protocol
  \[ u_i(t) = -\sum_{j \in \mathcal{N}_i} \langle \hat{x}_i(t) - \hat{x}_j(t) \rangle \]
- Measurement errors
  \[ e_i(t) = \hat{x}_i(t) - x_i(t) \]
- Closed-loop
  \[ \dot{x}(t) = -L\hat{x}(t) = -L(x(t) + e(t)) \]
- Disagreement
  \[ \delta(t) = x(t) - a1, \quad 1^T\delta(t) = 0 \]

**Event-based broadcasting**

- \( \hat{x}_i(t) = x_i(t_k), \ t \in [t_k^i, t_{k+1}^i] \)
- \( 0 \leq t_0^i \leq t_1^i \leq t_2^i \leq \cdots \)

### Trigger function for event-based control

**Trigger mechanism:** Define *trigger functions* \( f_i(\cdot) \) and trigger when

\[
f_i \left( t, x_i(t), \hat{x}_i(t), \bigcup_{j \in \mathcal{N}_i} \hat{x}_j(t) \right) > 0
\]

**Defines sequence of events:**

- \( t_{k+1}^i = \inf\{ t : t > t_k^i, f_i(t) > 0 \} \)

Find \( f_i \) such that

- \( |x_i(t) - x_j(t)| \to 0, \ t \to \infty \)
- no Zeno (no accumulation point in time)
- few inter-agent communications
Event-based control with constant thresholds

\[
\dot{x}(t) = u(t), \quad u(t) = -L\dot{x}(t) \tag{1}
\]

**Theorem (constant thresholds)**

Consider system (1) with undirected connected graph \( G \). Suppose that

\[
f_i(e_i(t)) = |e_i(t)| - c_0,
\]

with \( c_0 > 0 \). Then, for all \( x_0 \in \mathbb{R}^N \), the system does not exhibit Zeno behavior and for \( t \to \infty \),

\[
\|\delta(t)\| \leq \frac{\lambda_N(L)}{\lambda_2(L)} \sqrt{N}c_0.
\]

**Proof ideas:**
- Analytical solution of disagreement dynamics yields
  \[
  \|\delta(t)\| \leq e^{-\lambda_2(L)t} \|\delta(0)\| + \lambda_N(L) \int_0^t e^{-\lambda_2(L)(t-s)} \|\epsilon(s)\| ds
  \]
- Compute lower bound \( \tau \) on the inter-event intervals

Event-based control with exponentially decreasing thresholds

\[
\dot{x}(t) = u(t), \quad u(t) = -L\dot{x}(t) \tag{1}
\]

**Theorem (exponentially decreasing thresholds)**

Consider system (1) with undirected connected graph \( G \). Suppose that

\[
f_i(t, e_i(t)) = |e_i(t)| - c_1 e^{-\alpha t},
\]

with \( c_1 > 0 \) and \( 0 < \alpha < \lambda_2(L) \). Then, for all \( x_0 \in \mathbb{R}^N \), the system does not exhibit Zeno behavior and as \( t \to \infty \),

\[
\|\delta(t)\| \to 0.
\]

**Remarks**
- Asymptotic convergence: \( |x_i(t) - x_j(t)| \to 0, \ t \to \infty \)
- \( \lambda_2(L) \) is the rate of convergence for continuous-time consensus, so threshold need to decrease slower
Event-based control with exponentially decreasing thresholds and offset

\[ \dot{x}(t) = u(t), \quad u(t) = -L \dot{x}(t) \]  

(1)

Theorem (exponentially decreasing thresholds with offset)

Consider system (1) with undirected connected graph \( G \). Suppose that

\[ f_i(t, e_i(t)) = |e_i(t)| - (c_0 + c_1 e^{-\alpha t}), \]

with \( c_0, c_1 \geq 0 \), at least one positive, and \( 0 < \alpha < \lambda_2(L) \). Then, for all \( x_0 \in \mathbb{R}^N \), the system does not exhibit Zeno behavior and for \( t \to \infty \),

\[ \|\delta(t)\| \leq \frac{\lambda_N(L)}{\lambda_2(L)} \sqrt{N} c_0. \]

Example: threshold tuning

\[ \dot{x}(t) = -L \dot{x}(t) \]

\[ |e_i(t)| \leq c_0 + c_1 e^{-\alpha t} \]

\[ |e_i(t)| \leq 0.050 \]
Example: threshold tuning

\[ \dot{x}(t) = -L \dot{x}(t) \]
\[ |e_i(t)| \leq c_0 + c_1 e^{-\alpha t} \]
\[ |e_i(t)| \leq 0.050 \]

\[ |e_i(t)| \leq 0.001 \]
Example: threshold tuning

\begin{equation}
\dot{x}(t) = -L\dot{x}(t)
\end{equation}

\begin{equation}
|e_i(t)| \leq c_0 + c_1 e^{-\alpha t}
\end{equation}

\begin{equation}
|e_i(t)| \leq 0.001 + 0.249 e^{-0.9\lambda_2(L)t}
\end{equation}

\begin{equation}
|e_i(t)| \leq 0.250 e^{-0.9\lambda_2(L)t}
\end{equation}

Example: Event- vs time-triggered sampling

Example graphs showing comparison between time and event scheduling.

**Graph:**

**Sampling periods:**

- Time-scheduling:
  \( \tau_s = 0.350 \)
  \( \tau_{max} = 0.480 \)
- Event-scheduling:
  \( \tau_{mean} = 1.389 \)

**Notes:**

\( \tau_{max} \): largest stabilizing sampling period, see G. Xie et al., ACC2009
Example: Event- vs time-triggered sampling

Graph:

Sampling periods:
- Time-scheduling:
  \( \tau_s = 0.400 \)
  \( \tau_{\max} = 0.553 \)
- Event-scheduling:
  \( \tau_{\text{mean}} = 1.724 \)
Extension to double-integrator agents

**Multi-agent system model**
- $\dot{\xi}_i(t) = \zeta_i(t)$
- $\dot{\zeta}_i(t) = u_i(t)$
- Communication graph $G$

**Objective: Average consensus**
- $\xi_i(t) \xrightarrow{t \to \infty} \frac{1}{N} \sum_{i=1}^{N} \zeta_i(0) = b$
- $\xi_i(t) \xrightarrow{t \to \infty} \frac{1}{N} \sum_{i=1}^{N} \zeta_i(0) + bt$

**Consensus protocol**
- $u(t) = -L\xi(t) - \mu L\zeta(t)$

**Closed-loop dynamics**
$$\begin{bmatrix} \dot{\xi} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -L & -\mu L \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix}$$

Event-based implementation

**Multi-agent system model**
- $\dot{\xi}_i(t) = \zeta_i(t)$
- $\dot{\zeta}_i(t) = u_i(t)$
- Communication graph $G$

**Consensus protocol**
- $u(t) = -L\xi(t) - \mu L\zeta(t)$

**Event-based implemention**
- $u(t) = -L\left(\dot{\xi}(t) + \text{diag}(t - t_k^1, ..., t - t_k^N)\dot{\zeta}(t)\right) - \mu L\dot{\zeta}(t)$
- $\dot{\xi}_i(t) = \xi_i(t_k^1), \dot{\zeta}_i(t) = \zeta_i(t_k^1)$ for $t \in [t_k^i, t_{k+1}^i]$

**Measurement errors**
- $e_{\xi, i}(t) = (\dot{\xi}_i(t) + (t - t_k^i)\dot{\zeta}_i(t)) - \xi_i(t)$
- $e_{\zeta, i}(t) = \dot{\zeta}_i(t) - \zeta_i(t)$
Event-based control for double-integrator agents

\[ \dot{\xi}(t) = \zeta(t), \quad \dot{\zeta}(t) = u(t), \quad u(t) = -L \left( \dot{\xi}(t) + \text{diag}(t - t_k^1, \ldots, t - t_k^N)\dot{\zeta}(t) \right) - \mu L \dot{\zeta}(t) \tag{2} \]

**Theorem (double-integrator agents)**

Consider system (2) with undirected connected graph \( G \). Suppose that

\[ f_i(t, e_{\xi,i}(t), e_{\zeta,i}(t)) = \begin{bmatrix} e_{\xi,i}(t) \\ \mu e_{\zeta,i}(t) \end{bmatrix} - (c_0 + c_1 e^{-\alpha t}), \]

with \( c_0, c_1 \geq 0 \), at least one positive, and \( 0 < \alpha < |\Re(\lambda_3(G))| \). Then, for all \( \xi_0, \zeta_0 \in \mathbb{R}^N \), the system does not exhibit Zeno behavior and for \( t \to \infty \),

\[ \|\delta(t)\| \leq c_0 c_V \sqrt{\frac{\lambda_N(L)}{|\Re(\lambda_3(G))|}} \sqrt{2N}. \]

**Example:** Formation control for non-holonomic mobile robots

- ![Graph](image)
- ![Graph](image)
- ![Graph](image)

- Uses feedback linearization and the double-integrator control strategy
Summary
• Multi-agent control under event-based communication
• Guaranteed convergence and well-posedness

Extensions
• How to estimate $\lambda_2(L)$ in a distributed way?
  – See Aragues et al. (ACC, 2012)
• How to handle general agent dynamics?
  – See Guinaldo et al. (IET, 2013)
• How to handle network delays and packet losses?
  – See Guinaldo et al. (CDC, 2012)

Lecture 12 Outline
• Distributed event-based control
• Anti-windup for event-based control
• Event-based PID control
Event-triggered PI control

Event-triggered PI control with saturation
Event-triggered PI control with saturation

- Industrial applications are generally affected by actuator limitations.

1. Does actuator saturation affect event-triggered PI control?
2. Under what conditions can we guarantee stability?
3. How to overcome potential effects of actuator saturation?

Example

- Plant:
  \[
  \dot{x}(t) = 0.1x(t) + \tilde{u}(t) + 0.1d(t), \quad x(0) = 0 \\
  y(t) = x(t)
  \]

- Exogenous signals:
  \[
  w(t) = \bar{w} = 1.5 \\
  d(t) = \bar{d} = 0.1
  \]

- Actuator saturation:
  \[
  \tilde{u}(t) = \begin{cases} 
  0.4, & \text{for } u(t) > 0.4; \\
  u(t), & \text{for } -0.4 \leq u(t) \leq 0.4; \\
  -0.4, & \text{for } u(t) < -0.4;
  \end{cases}
  \]

- PI controller
  \[
  \dot{x}_1(t) = y(t) - w(t), \quad x_1(0) = 0 \\
  u(t) = -x_1(t) - 1.6(y(t) - w(t))
  \]

Lehmann, Johansson: Event-triggered PI control subject to actuator saturation. IFAC PID conference, 2012
Example: Without saturation

Event generator invokes a sensor transmission whenever output error reach a predefined fixed threshold:

\[ |e(t)| = |y(t) - y(t_k)| = \varepsilon. \]
Motivating example

Need to take saturation and wind-up into account when designing event-based control systems.
Mathematical model

- **Plant:**
  \[ \dot{x}(t) = Ax(t) + Bu(t) + Ed(t), \quad x(0) = x_0 \]
  \[ \tilde{u}(t) = \text{sat}(u(t)) \]
  \[ \text{sat}(u_i(t)) = \begin{cases} 
  u_0, & \text{for } u_i(t) > u_0 \\
  u_i(t), & \text{for } -u_0 \leq u(t) \leq u_0 \\
  -u_0, & \text{for } u_i(t) < -u_0
  \end{cases} \quad \forall i \in \{1, 2, \ldots, m\} \]

- **Event generator:** \[ \|x(t) - x(t_k)\| = \bar{e} \]

- **PI controller:**
  \[ \dot{x}_1(t) = x(t) - e(t) - w(t), \quad x_1(0) = x_0 \]
  \[ u(t) = K_1x_1(t) + K_P(x(t) - e(t) - w(t)) \]

- **State error:** \[ e(t) = x(t) - x(t_k) \]

- For the sake of simplicity: \[ w(t) = d(t) = 0 \]

---

**Augmented state vector:**

\[ x_{\text{a}}(t) = \begin{pmatrix} x(t) \\ x_1(t) \end{pmatrix} \]

**State-space model of the event-triggered PI-control loop:**

\[ \dot{x}_{\text{a}}(t) = A_1x_{\text{a}}(t) + B_1\text{sat}(K_1x_1(t) + K_P(x(t) - e(t))) - F_1e(t) \]
\[ x_{\text{a}}(0) = x_{\text{a}0} \]
Transformation of saturation nonlinearity

\[ \psi(u) = \text{sat}(u) - u \]

Transformed state-space model of the event-triggered PI-control loop:

\[
\begin{align*}
\dot{x}_a(t) &= \bar{A}_1 x_a(t) + B_1 \psi(K x_a(t) - K_P e(t)) - F_1 e(t) \\
x_a(0) &= x_{a0}
\end{align*}
\]

\[
\bar{A}_1 = \begin{pmatrix} A + BK_p & BK_1 \\ I & O \end{pmatrix}; B_1 = \begin{pmatrix} B \\ O \end{pmatrix}; F_1 = \begin{pmatrix} BK_p \\ I \end{pmatrix}; K = \begin{pmatrix} K_p & K_1 \end{pmatrix}
\]

Nonlinearity transformation enables tighter stability conditions [Tarbourieh et al, 2006]

---

**Theorem: Region of stability**

If there exist a symmetric positive definite matrix \( W \), a positive definite diagonal matrix \( S \), a matrix \( Z \), a positive scalar \( \eta \) and two a priori fixed positive scalars \( \tau_1 \) and \( \tau_2 \) satisfying

\[
\begin{pmatrix} W A_1^T + \bar{A}_1 W + \tau_1 W & B_1 S - W K^T - Z^T & -F_1 \\ * & -2S & -K_P \\ * & * & -\tau_2 R \end{pmatrix} < 0
\]

\[-\tau_1 \delta + \tau_2 \eta < 0\]

\[
\begin{pmatrix} W & Z^T \\ * & \eta u_0^2 \end{pmatrix} \geq 0, i \in 1, ..., m
\]

then for \( e \in \mathcal{W} = \{ e : e^T Re = \delta^{-1} \} \) \((R = I, \delta^{-1} = \epsilon^2)\) the ellipsoid \( \mathcal{E} = \{ x_n : x_n^T P x_n = \eta^{-1} \} \), with \( P = W^{-1} \), is a region of stability.

- Computational tool to estimate region of stability for saturated event-based control
- Extends results for continuous-time systems [Tarbourieh; Zaccarian & Teel, 2011]
Example revisited

- **Plant:**
  \[
  \dot{x}(t) = 0.1x(t) + \bar{u}(t) + 0.1d(t), \quad x(0) = 0
  
  y(t) = x(t)
  \]

- **Exogenous signals:**
  \[
  w(t) = \bar{w} = 1.5
  
  d(t) = \bar{d} = 0.1
  \]

- **Actuator saturation:**
  \[
  \tilde{u}(t) = \begin{cases} 
  0.4, & \text{for } u(t) > 0.4; \\
  u(t), & \text{for } -0.4 \leq u(t) \leq 0.4; \\
  -0.4, & \text{for } u(t) < -0.4;
  \end{cases}
  \]

- **PI controller**
  \[
  \dot{x}_1(t) = y(t) - w(t), \quad x_1(0) = 0
  
  u(t) = -x_1(t) - 1.6(y(t) - w(t))
  \]

Stability regions

- **Stability regions diagram**
  - \(\tau = 0\) (CT control)
  - \(\tau = 0.15\)
  - \(\tau = 1.05\)
Stable trajectory

Unstable trajectory
Anti-windup for event-based control

Anti-windup schemes for conventional control systems, e.g., Åström & Hägglund, 1995
Anti-windup for event-based PI control

- Adapted dynamics of the controller state:
  \[ \dot{x}_1(t) = x_1(t) - e(t) - w(t) + K_{sat} \phi(u), \quad x_1(0) = x_{10} \]

- Transformed state-space model of the event-triggered PI-control loop:
  \[
  \dot{x}_a(t) = \bar{A}_1 x_a(t) + B_1 \phi(K x_a(t) - K_P e(t)) - F_1 e(t), \quad x_a(0) = x_{a0}
  \]
  \[
  \bar{A}_1 = \begin{pmatrix}
  A + BK_P & BK_I \\
  I & O
  \end{pmatrix};
  B_1 = \begin{pmatrix}
  B \\
  K_{sat}
  \end{pmatrix};
  F_1 = \begin{pmatrix}
  BK_P \\
  I
  \end{pmatrix};
  K = \begin{pmatrix}
  K_P & K_I
  \end{pmatrix}
  \]

Example: Stability regions with anti-windup
System evolution with anti-windup

Alternative event-based anti-windup scheme

$|u(t) - u(t_k)| = e_c$ if $\dot{u}(t) - u(t) \neq 0$
Summary
• Actuator saturation significantly affects event-based control
• Region of stability computable by means of LMIs.
• Anti-windup improves both the behavior of the event-based control loop and the size of the region of stability

Extensions
• General controller dynamics and other triggering rules
  – See Kiener et al. (DEDS, 2013)
• Many possibilities for combining event-based signaling with various feedback control schemes
  – Other anti-windup schemes, feedforward control etc

Lecture 12 Outline
• Distributed event-based control
• Anti-windup for event-based control
• Event-based PID control
Proportion-Integral-Derivative control

\[ u_t = K_P e_t + K_I \int e_s ds + K_D \frac{de_t}{dt} \]

How extend PID control to event-based control?

Event-detector for PID control

Rabi and J., WICON, 2008
Control generator for PID control

Control alphabet consists of three symbols, which are activated depending on the event.

Rabi and J., WICON, 2008

Example: Integral control

Communicate only when integral error triggers events.
Lecture 12 Outline

• Distributed event-based control
• Anti-windup for event-based control
• Event-based PID control