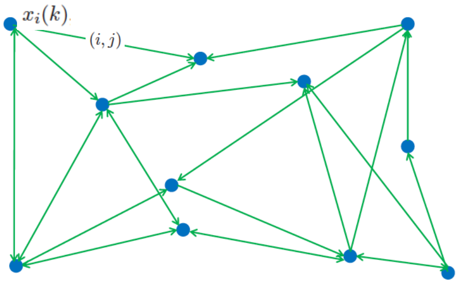
Lecture 12: Distributed and saturated event-based control

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

$$\dot{x}_i(t) = u_i(t)$$

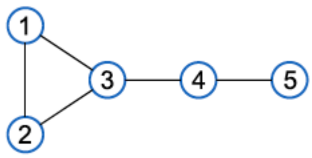
- Communication graph G
  A: undirected, connected

Adjacency matrix A with $a_{ij} = 1$ 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

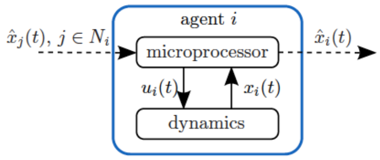
$$\dot{x}(t) = -Lx(t)$$

with Laplacian matrix $L = D - A$

Event-based implementation?

Event-based average consensus

Event-based scheduling of measurement broadcasts:



Event-based broadcasting

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

- Consensus protocol

$$u_i(t) = -\sum_{j \in N_i} (\hat{x}_i(t) - \hat{x}_j(t))$$

- 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

$$\delta(t) = x(t) - a\mathbf{1}, \qquad \mathbf{1}^T\delta(t) \equiv 0$$

Trigger function for event-based control

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

$$f_i\left(t, x_i(t), \hat{x}_i(t), \bigcup_{j \in N_i} \hat{x}_j(t)\right) > 0$$

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

$$\dot{x}(t) = u(t), \qquad u(t) = -L\hat{x}(t) \qquad (1)$$

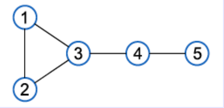
Theorem (constant thresholds)

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

$$\|\delta(t)\| \leq e^{-\lambda_2(L)t}\|\delta(0)\| + \lambda_N(L) \int_0^t e^{-\lambda_2(L)(t-s)}\|e(s)\|ds$$

- Compute lower bound τ on the inter-event intervals

Event-based control with exponentially decreasing thresholds

$$\dot{x}(t) = u(t), \qquad u(t) = -L\hat{x}(t) \qquad (1)$$

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

$$\dot{x}(t) = u(t), \qquad u(t) = -L\hat{x}(t) \qquad (1)$$

Theorem (exponentially decreasing thresholds with offset)

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

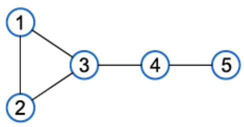
$$f_i(t, e_i(t)) = |e_i(t)| - (c_0 + c_1 e^{-\alpha t}),$$

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 $t \to \infty$,

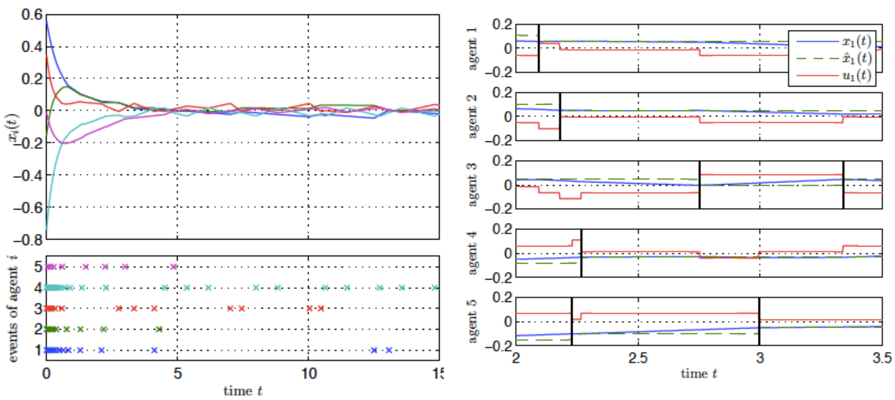
$$\|\delta(t)\| \leq \frac{\lambda_N(L)}{\lambda_2(L)} \sqrt{N} c_0.$$

Example: threshold tuning

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

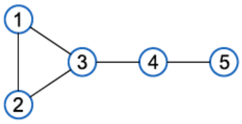


$$|e_i(t)| \leq 0.050$$

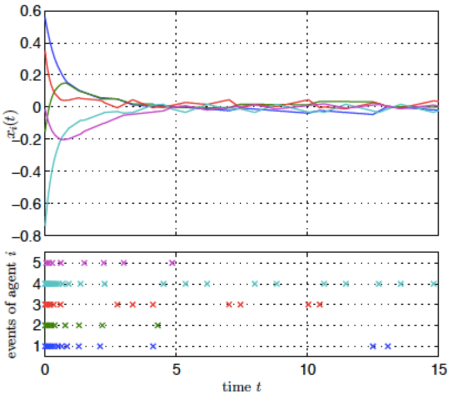


Example: threshold tuning

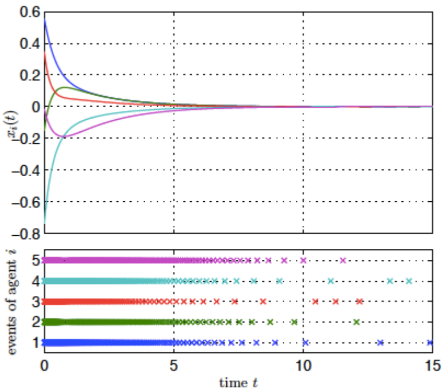
$$\dot{x}(t) = -L\hat{x}(t)$$
$$|e_i(t)| \leq c_0 + c_1 e^{-\alpha t}$$



$|e_i(t)| \leq 0.050$

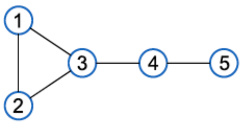


$|e_i(t)| \leq 0.001$

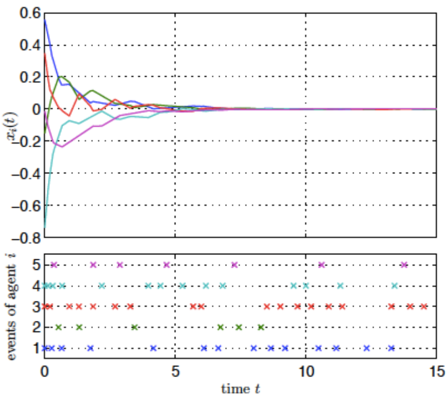


Example: threshold tuning

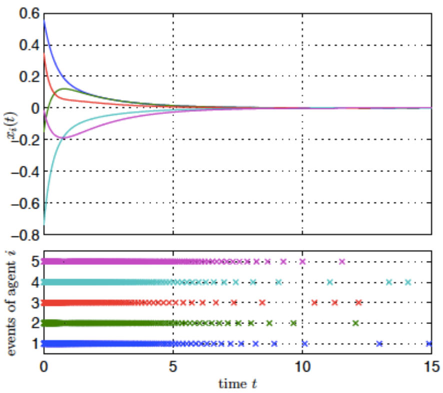
$$\dot{x}(t) = -L\hat{x}(t)$$
$$|e_i(t)| \leq c_0 + c_1 e^{-\alpha t}$$



$|e_i(t)| \leq 0.001 + 0.249\, e^{-0.9\lambda_2(L)t}$



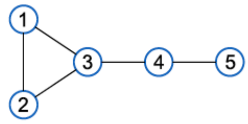
$|e_i(t)| \leq 0.001$



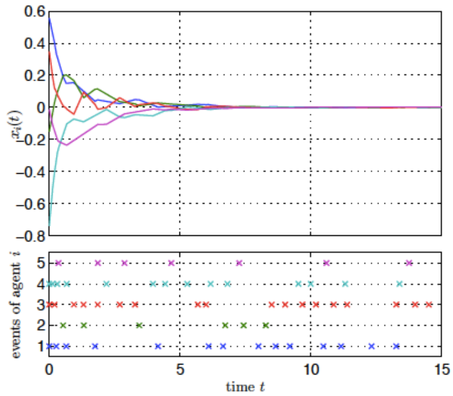
Example: threshold tuning

$\dot{x}(t) = -L\hat{x}(t)$

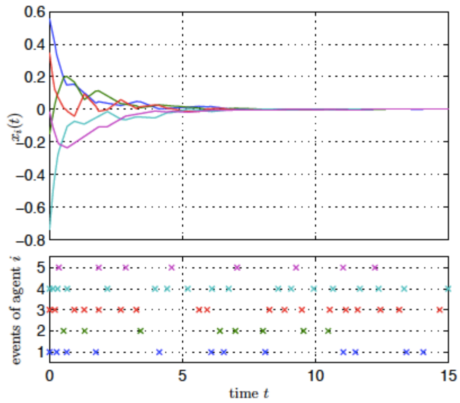
$|e_i(t)| \le c_0 + c_1 e^{-\alpha t}$



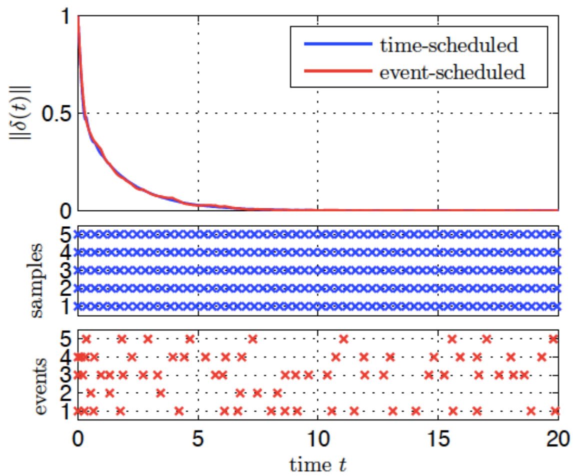
$|e_i(t)| \le 0.001 + 0.249\, e^{-0.9\lambda_2(L)t}$



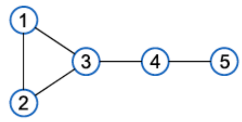
$|e_i(t)| \le 0.250\, e^{-0.9\lambda_2(L)t}$



Example: Event- vs time-triggered sampling



Graph:

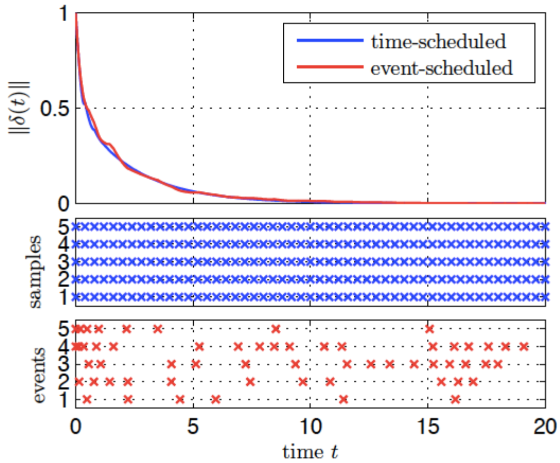


Sampling periods:

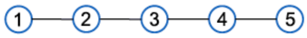
- Time-scheduling:
  $\tau_s = 0.350$
  $\tau_{max} = 0.480$
- Event-scheduling:
  $\tau_{mean} = 1.389$

τ_{max} : largest stabilizing sampling period, see *G. Xie et al., ACC2009*

Example: Event- vs time-triggered sampling



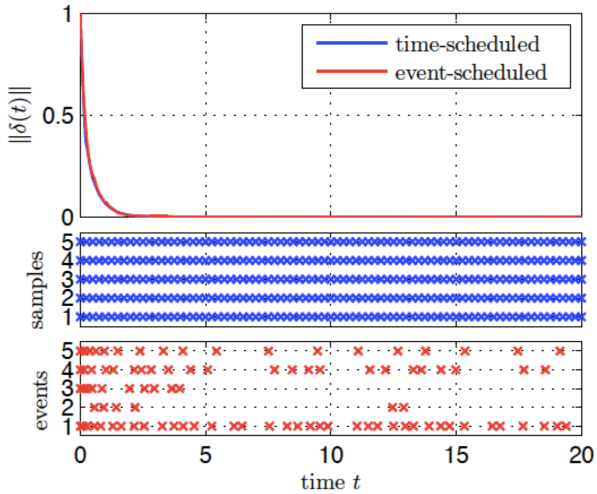
Graph:



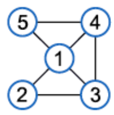
Sampling periods:

- Time-scheduling:
  $\tau_s = 0.400$
  $\tau_{max} = 0.553$
- Event-scheduling:
  $\tau_{mean} = 1.724$

Example: Event- vs time-triggered sampling



Graph:



Sampling periods:

- Time-scheduling:
  $\tau_s = 0.250$
  $\tau_{max} = 0.400$
- Event-scheduling:
  $\tau_{mean} = 1.053$

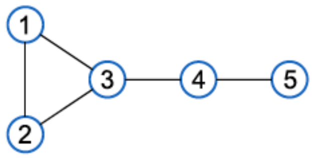
Extension to double-integrator agents

Multi-agent system model

- $\dot{\xi}_i(t) = \zeta_i(t)$
  $\dot{\zeta}_i(t) = u_i(t)$
- communication graph G

Objective: Average consensus

$$\zeta_i(t) \xrightarrow{t\to\infty} \frac{1}{N}\sum_{i=1}^{N} \zeta_i(0) = b$$
$$\xi_i(t) \xrightarrow{t\to\infty} \frac{1}{N}\sum_{i=1}^{N} \xi_i(0) + bt$$



Consensus protocol

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

Closed-loop dynamics

$$\begin{bmatrix} \dot{\xi} \\ \dot{\zeta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -L & -\mu L \end{bmatrix}}_{\Gamma} \begin{bmatrix} \xi \\ \zeta \end{bmatrix}$$

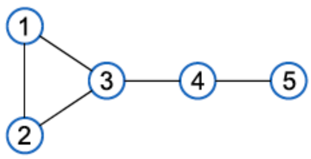
Event-based implementation

Multi-agent system model

- $\dot{\xi}_i(t) = \zeta_i(t)$
  $\dot{\zeta}_i(t) = u_i(t)$
- communication graph G

Consensus protocol

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



$$u(t) = -L\left(\hat{\xi}(t) + \text{diag}(t - t_k^1, \ldots, t - t_k^N)\hat{\zeta}(t)\right) - \mu L\hat{\zeta}(t)$$

$$\hat{\xi}_i(t) = \xi_i(t_k^i),\ \hat{\zeta}_i(t) = \zeta_i(t_k^i) \text{ for } t \in [t_k^i, t_{k+1}^i[$$

Measurement errors

- $e_{\xi,i}(t) = (\hat{\xi}_i(t) + (t - t_k^i)\hat{\zeta}_i(t)) - \xi_i(t)$
- $e_{\zeta,i}(t) = \hat{\zeta}_i(t) - \zeta_i(t)$

Event-based control for double-integrator agents

$$\begin{array}{l}\dot{\xi}(t) = \zeta(t) \\ \dot{\zeta}(t) = u(t)\end{array}, \quad u(t) = -L\left(\hat{\xi}(t) + \text{diag}(t - t_k^1, ..., t - t_k^N)\hat{\zeta}(t)\right) - \mu L \hat{\zeta}(t) \quad (2)$$

Theorem (double-integrator agents)

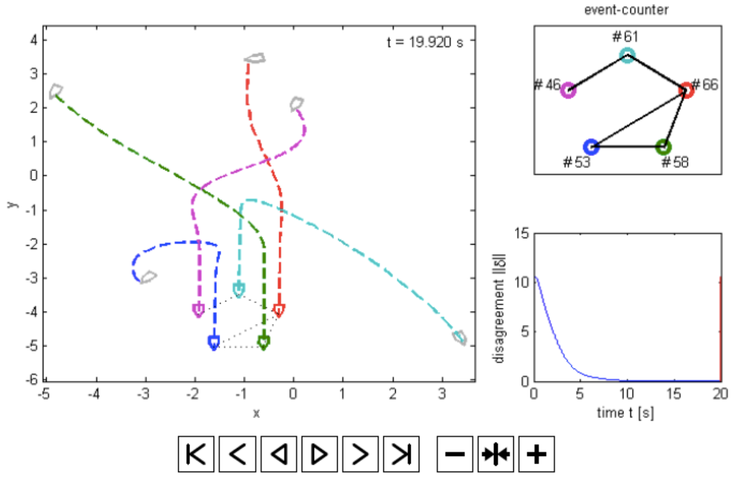
Consider system (2) with undirected connected graph G. Suppose that

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

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

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

Example: Formation control for non-holonomic mobile robots



- uses feedback linearization and the double-integrator control strategy

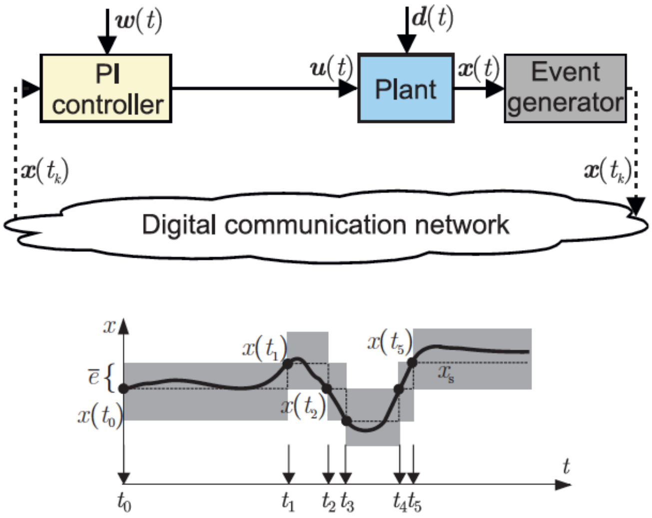
Summary
- Multi-agent control under event-based communication
- Guaranteed convergence and well-posedness

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)

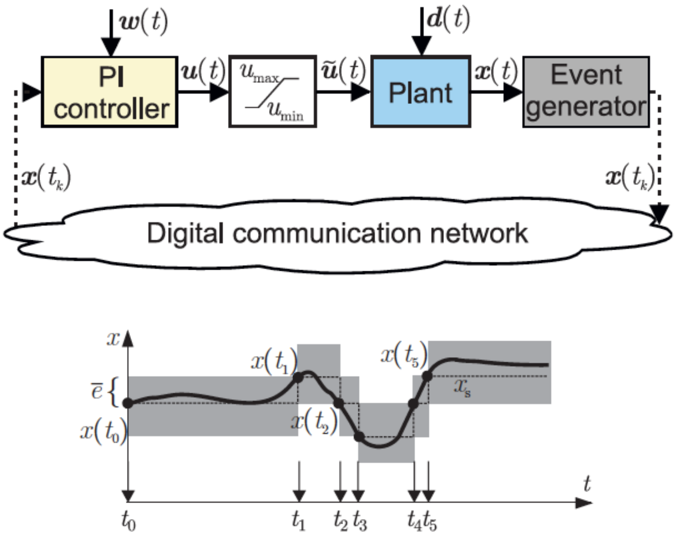
Lecture 12 Outline

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

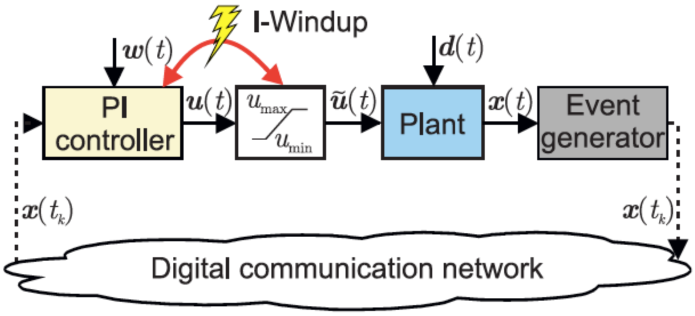
Event-triggered PI control



Event-triggered PI control with saturation



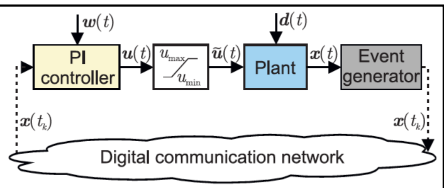
Event-triggered PI control with saturation



- 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:

$$\begin{aligned} \dot{x}(t) &= 0.1x(t) + \tilde{u}(t) + 0.1d(t), \quad x(0) = 0 \\ y(t) &= x(t) \end{aligned}$$

- Exogenous signals:

$$\begin{aligned} w(t) &= \bar{w} = 1.5 \\ d(t) &= \bar{d} = 0.1 \end{aligned}$$

- Actuator saturation:

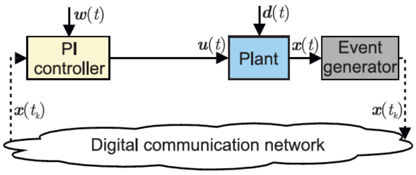
$$\tilde{u}(t) = \begin{cases} 0.4, & \text{for } u(t) > 0.4; \\ u(t), & \text{for } -0.4 \le u(t) \le 0.4; \\ -0.4, & \text{for } u(t) < -0.4; \end{cases}$$

- PI controller

$$\begin{aligned} \dot{x}_I(t) &= y(t) - w(t), \quad x_I(0) = 0 \\ u(t) &= -x_I(t) - 1.6(y(t) - w(t)) \end{aligned}$$

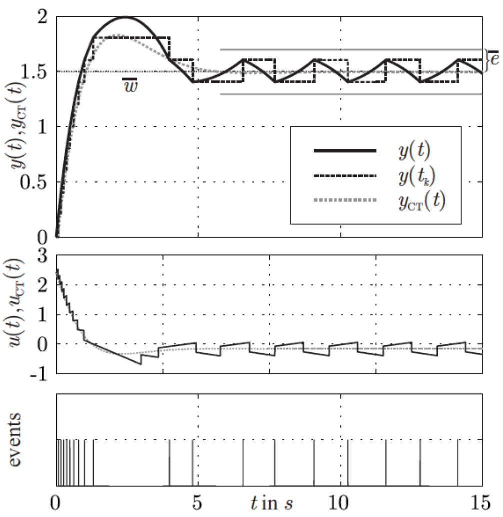
Lehmann, Johansson: Event-triggered PI control subject to actuator saturation. *IFAC PID conference*, 2012

Example: Without saturation

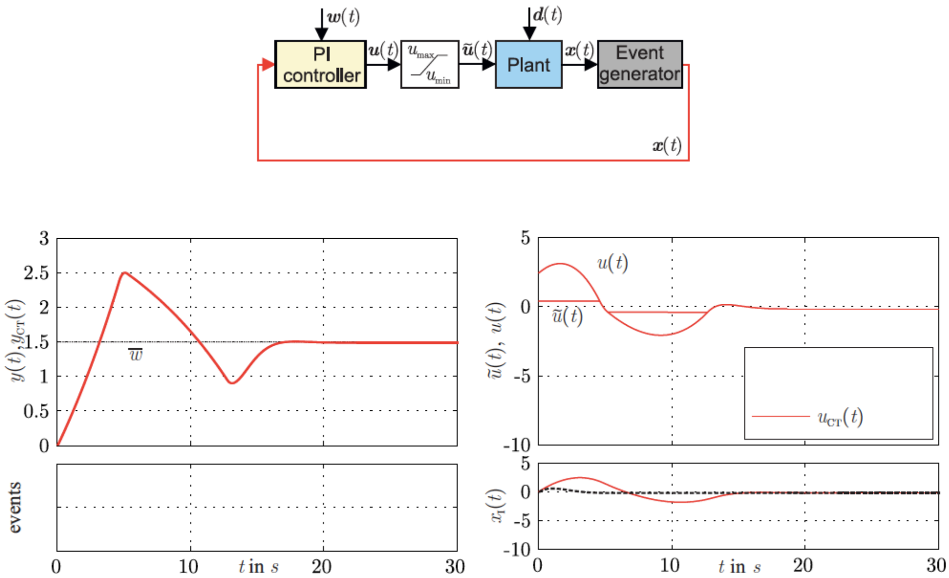


Event generator invokes a sensor transmission whenever output error reach a predefined fixed threshold:

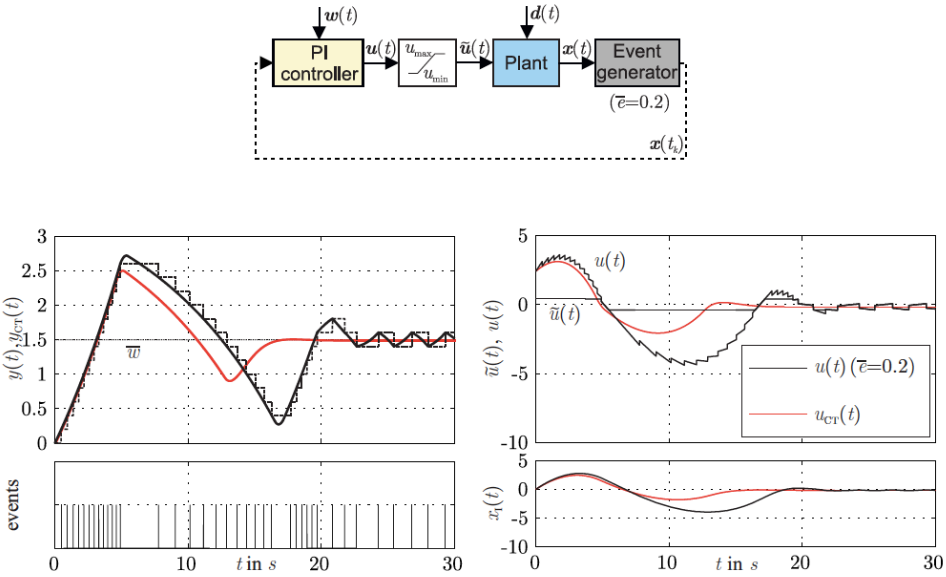
$$|e(t)| = |y(t) - y(t_k)| = \bar{e}.$$



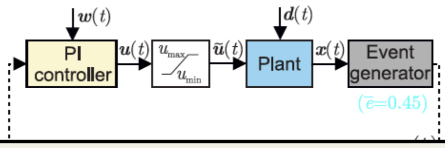
Motivating example



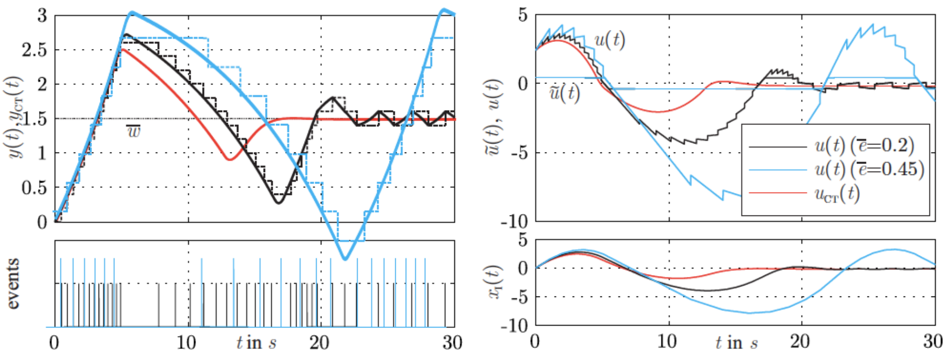
Motivating example



Motivating example



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



Mathematical model

- Plant:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\tilde{\boldsymbol{u}}(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \tilde{\boldsymbol{u}}(t) &= \mathbf{sat}(\boldsymbol{u}(t)) \\ \text{sat}(u_i(t)) &= \begin{cases} u_0, & \text{for } u_i(t) > u_0 \\ u_i(t), & \text{for } -u_0 \le u(t) \le u_0 \quad \forall i \in \{1,2,..,m\} \\ -u_0, & \text{for } u_i(t) < -u_0 \end{cases}\end{aligned}$$

- Event generator: $\|\boldsymbol{x}(t) - \boldsymbol{x}(t_k)\| = \bar{e}$
- PI controller:

$$\begin{aligned}\dot{\boldsymbol{x}}_{\mathrm{I}}(t) &= \boldsymbol{x}(t) - \boldsymbol{e}(t) - \boldsymbol{w}(t), \quad \boldsymbol{x}_{\mathrm{I}}(0) = \boldsymbol{x}_0 \\ \boldsymbol{u}(t) &= \boldsymbol{K}_{\mathrm{I}}\boldsymbol{x}_{\mathrm{I}}(t) + \boldsymbol{K}_{\mathrm{P}}(\boldsymbol{x}(t) - \boldsymbol{e}(t) - \boldsymbol{w}(t))\end{aligned}$$

- **State error:** $\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{x}(t_k)$
- For the sake of simplicity: $\boldsymbol{w}(t) = \boldsymbol{d}(t) = \mathbf{0}$

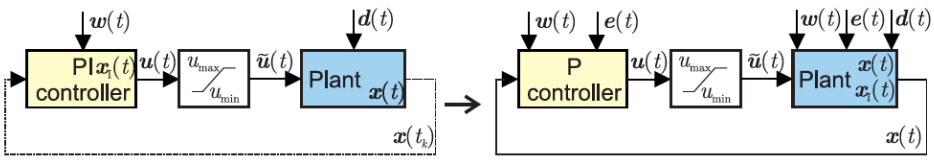
Mathematical model

Augmented state vector:

$$\boldsymbol{x}_{\mathrm{a}}(t) = \begin{pmatrix} \boldsymbol{x}(t) \\ \boldsymbol{x}_{\mathrm{I}}(t) \end{pmatrix}$$

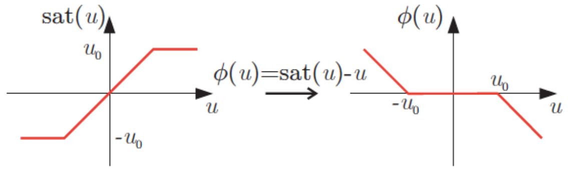
State-space model of the event-triggered PI-control loop:

$$\begin{aligned}\dot{\boldsymbol{x}}_{\mathrm{a}}(t) &= \boldsymbol{A}_{\mathrm{I}}\boldsymbol{x}_{\mathrm{a}}(t) + \boldsymbol{B}_{\mathrm{I}}\mathbf{sat}(\boldsymbol{K}_{\mathrm{I}}\boldsymbol{x}_{\mathrm{I}}(t) + \boldsymbol{K}_{\mathrm{P}}(\boldsymbol{x}(t) - \boldsymbol{e}(t))) - \boldsymbol{F}_{\mathrm{I}}\boldsymbol{e}(t) \\ \boldsymbol{x}_{\mathrm{a}}(0) &= \boldsymbol{x}_{\mathrm{a}0}\end{aligned}$$



Transformation of saturation nonlinearity

$$\phi(\boldsymbol{u}) = \mathrm{sat}(\boldsymbol{u}) - \boldsymbol{u}$$



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

$$\begin{aligned} \dot{\boldsymbol{x}}_{\mathrm{a}}(t) &= \bar{\boldsymbol{A}}_{\mathrm{I}}\boldsymbol{x}_{\mathrm{a}}(t) + \boldsymbol{B}_{\mathrm{I}}\phi(\boldsymbol{K}\boldsymbol{x}_{\mathrm{a}}(t) - \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}(t)) - \boldsymbol{F}_{\mathrm{I}}\boldsymbol{e}(t) \\ \boldsymbol{x}_{\mathrm{a}}(0) &= \boldsymbol{x}_{\mathrm{a}0} \end{aligned}$$

$$\bar{\boldsymbol{A}}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\mathrm{P}} & \boldsymbol{B}\boldsymbol{K}_{\mathrm{I}} \\ \boldsymbol{I} & \boldsymbol{O} \end{pmatrix}; \boldsymbol{B}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{O} \end{pmatrix}; \boldsymbol{F}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{B}\boldsymbol{K}_{\mathrm{P}} \\ \boldsymbol{I} \end{pmatrix}; \boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}_{\mathrm{P}} & \boldsymbol{K}_{\mathrm{I}} \end{pmatrix}$$

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

Theorem: Region of stability

If there exist a symmetric positive definite matrix $\boldsymbol{W}$, a positive definite diagonal matrix $\boldsymbol{S}$, a matrix $\boldsymbol{Z}$, a positive scalar η and two a priori fixed positive scalars τ_1 and τ_2 satisfying

$$\begin{bmatrix} \boldsymbol{W}\bar{\boldsymbol{A}}_{\mathrm{I}}^T + \bar{\boldsymbol{A}}_{\mathrm{I}}\boldsymbol{W} + \tau_1\boldsymbol{W} & \boldsymbol{B}_{\mathrm{I}}\boldsymbol{S} - \boldsymbol{W}\boldsymbol{K}^T - \boldsymbol{Z}^T & -\boldsymbol{F}_{\mathrm{I}} \\ \star & -2\boldsymbol{S} & -\boldsymbol{K}_{\mathrm{P}} \\ \star & \star & -\tau_2\boldsymbol{R} \end{bmatrix} < 0$$

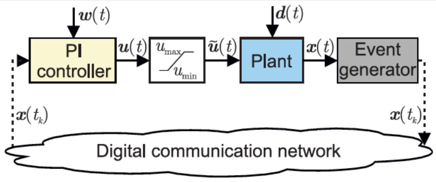
$$-\tau_1\delta + \tau_2\eta < 0$$

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{Z}_i^T \\ \star & \eta u_0^2 \end{bmatrix} \geq 0, i \in 1, ..., m$$

then for $\boldsymbol{e} \in \mathcal{W} = \{\boldsymbol{e} : \boldsymbol{e}^T\boldsymbol{R}\boldsymbol{e} = \delta^{-1}\}$ $(\boldsymbol{R} = \boldsymbol{I}, \delta^{-1} = \bar{e}^2)$ the ellipsoid $\mathcal{E} = \{\boldsymbol{x}_{\mathrm{a}} : \boldsymbol{x}_{\mathrm{a}}^T\boldsymbol{P}\boldsymbol{x}_{\mathrm{a}} = \eta^{-1}\}$, with $\boldsymbol{P} = \boldsymbol{W}^{-1}$, is a region of stability.

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

Example revisited



- Plant:

$$\begin{aligned}\dot{x}(t) &= 0.1x(t) + \tilde{u}(t) + 0.1d(t), \quad x(0) = 0\\ y(t) &= x(t)\end{aligned}$$

- Exogenous signals:

$$\begin{aligned}w(t) &= \bar{w} = 1.5\\ d(t) &= \bar{d} = 0.1\end{aligned}$$

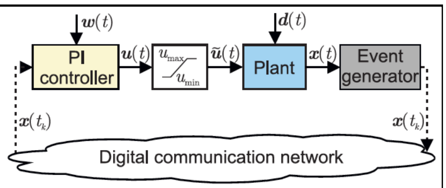
- Actuator saturation:

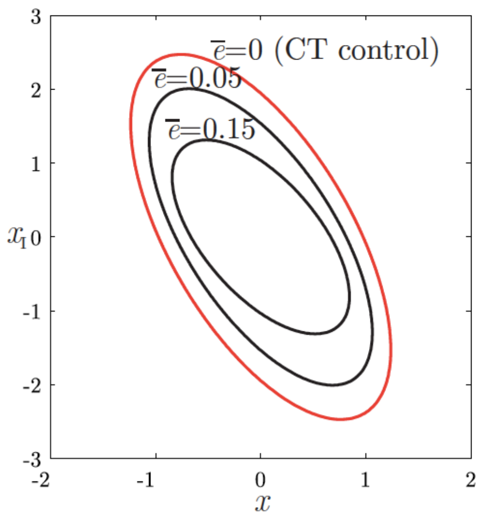
$$\tilde{u}(t) = \begin{cases} 0.4, & \text{for } u(t) > 0.4;\\ u(t), & \text{for } -0.4 \le u(t) \le 0.4;\\ -0.4, & \text{for } u(t) < -0.4;\end{cases}$$

- PI controller

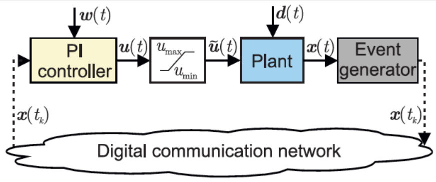
$$\begin{aligned}\dot{x}_{\mathrm{I}}(t) &= y(t) - w(t), \quad x_{\mathrm{I}}(0) = 0\\ u(t) &= -x_{\mathrm{I}}(t) - 1.6(y(t) - w(t))\end{aligned}$$

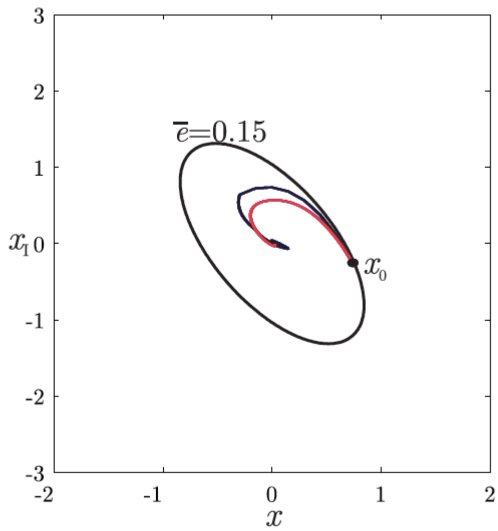
Stability regions



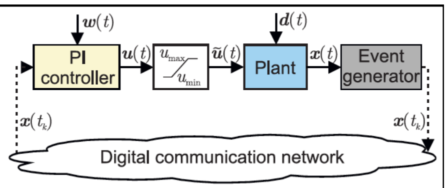


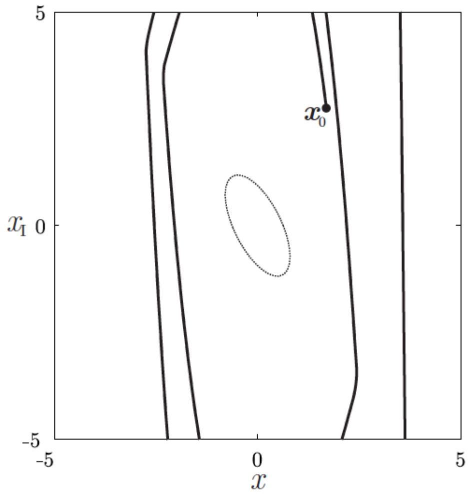
Stable trajectory



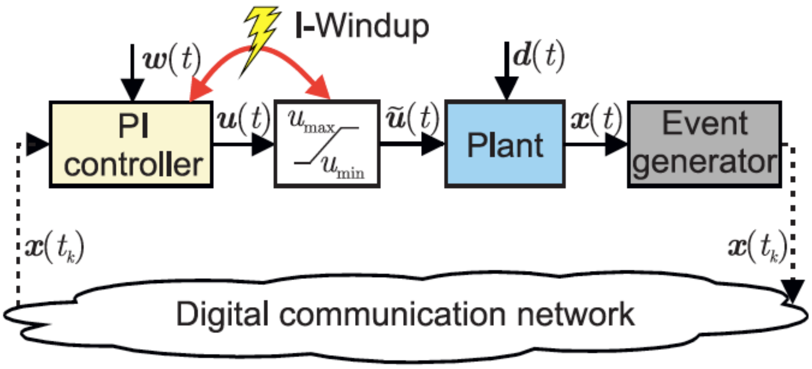


Unstable trajectory

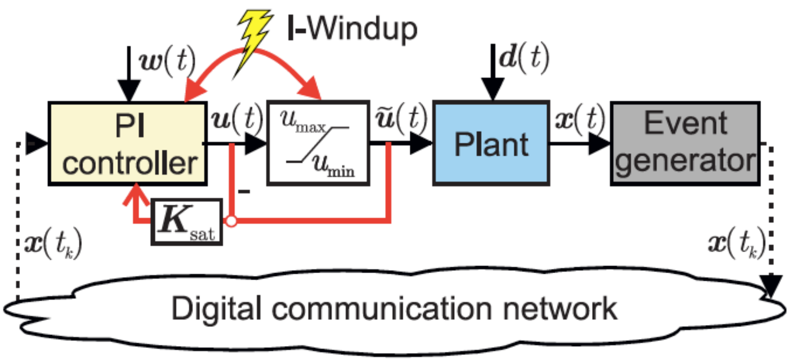




Anti-windup for event-based control

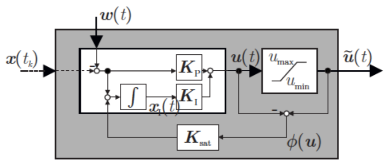


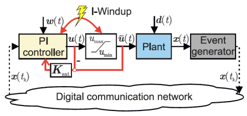
Anti-windup for event-based control



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:

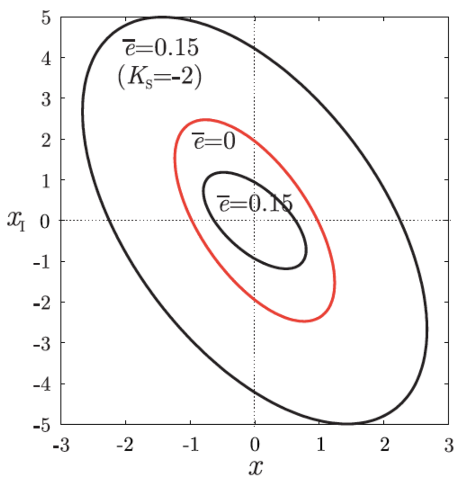
$$\dot{\boldsymbol{x}}_{\mathrm{I}}(t) = \boldsymbol{x}(t) - \boldsymbol{e}(t) - \boldsymbol{w}(t) + \boldsymbol{K}_{\mathrm{sat}}\phi(\boldsymbol{u}), \quad \boldsymbol{x}_{\mathrm{I}}(0) = \boldsymbol{x}_{\mathrm{I0}}$$

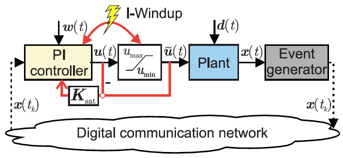
- Transformed state-space model of the event-triggered PI-control loop:

$$\dot{\boldsymbol{x}}_{\mathrm{a}}(t) = \bar{\boldsymbol{A}}_{\mathrm{I}}\boldsymbol{x}_{\mathrm{a}}(t) + \boldsymbol{B}_{\mathrm{I}}\phi(\boldsymbol{K}\boldsymbol{x}_{\mathrm{a}}(t) - \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}(t)) - \boldsymbol{F}_{\mathrm{I}}\boldsymbol{e}(t), \quad \boldsymbol{x}_{\mathrm{a}}(0) = \boldsymbol{x}_{\mathrm{a0}}$$

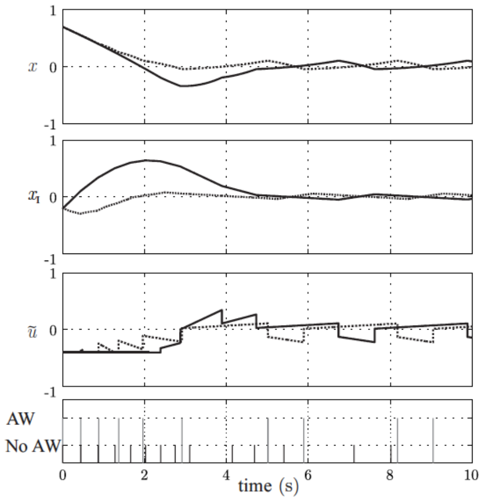
$$\bar{\boldsymbol{A}}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\mathrm{P}} & \boldsymbol{B}\boldsymbol{K}_{\mathrm{I}} \\ \boldsymbol{I} & \boldsymbol{O} \end{pmatrix}; \boldsymbol{B}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{K}_{\mathrm{sat}} \end{pmatrix}; \boldsymbol{F}_{\mathrm{I}} = \begin{pmatrix} \boldsymbol{B}\boldsymbol{K}_{\mathrm{P}} \\ \boldsymbol{I} \end{pmatrix}; \boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}_{\mathrm{P}} & \boldsymbol{K}_{\mathrm{I}} \end{pmatrix}$$

Example: Stability regions with anti-windup

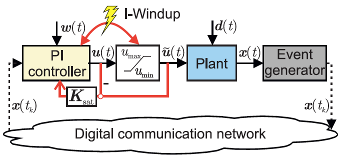




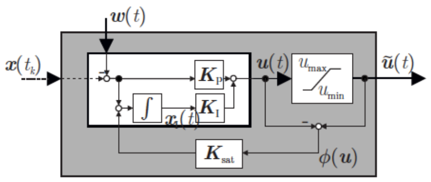
System evolution with anti-windup

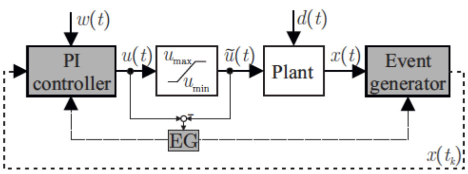


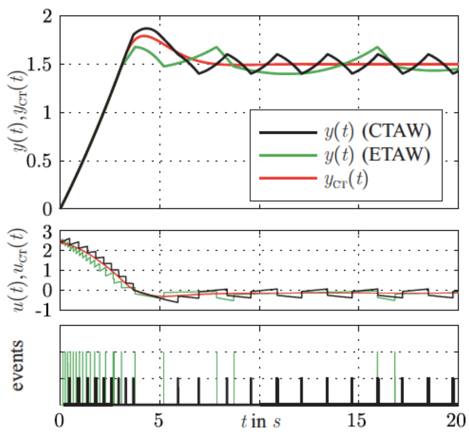
(solid line: no anti windup, dotted line: anti windup)



Alternative event-based anti-windup scheme







$$|u(t) - u(t_k)| = \bar{e}_c \quad \text{if} \quad \tilde{u}(t) - u(t) \neq 0$$

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

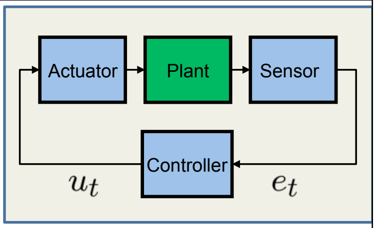
- General controller dynamics and other triggering rules
  - See Kiener et al. (DEDS, 2013)
- Many possibilities for combining event-based signaling with various feedback control schemes
  - Other anti-windup schemes, feedforward control etc

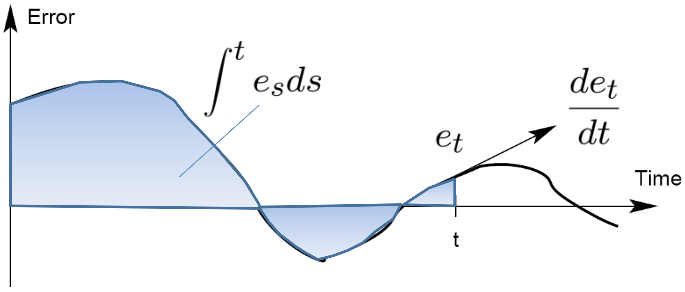
Lecture 12 Outline

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

Proportion-Integral-Derivative control

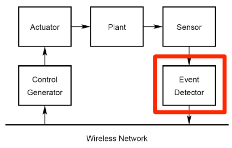
$$u_t = K_P e_t + K_I \int^t e_s ds + K_D \frac{de_t}{dt}$$

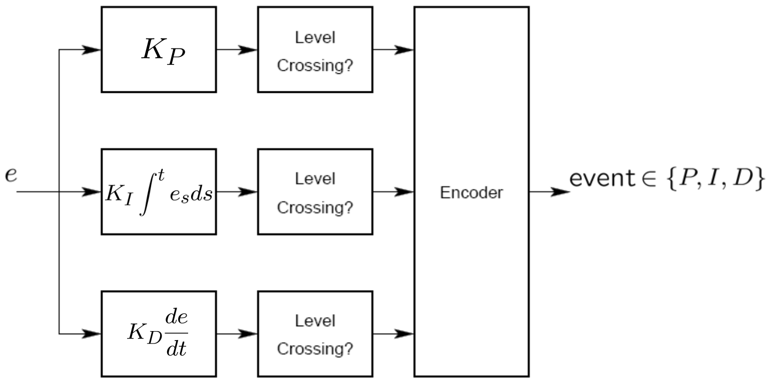




How extend PID control to event-based control?

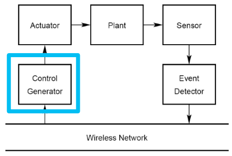
Event-detector for PID control

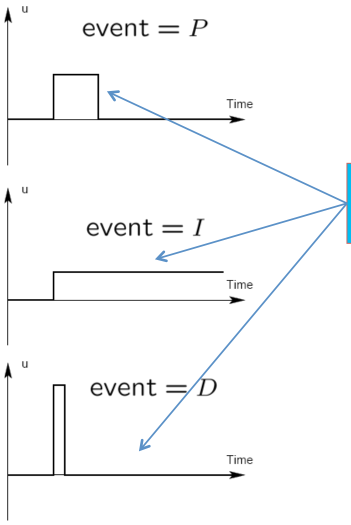




Rabi and J., *WICON, 2008*

Control generator for PID control

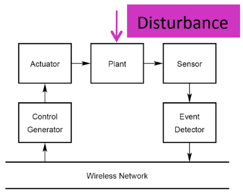


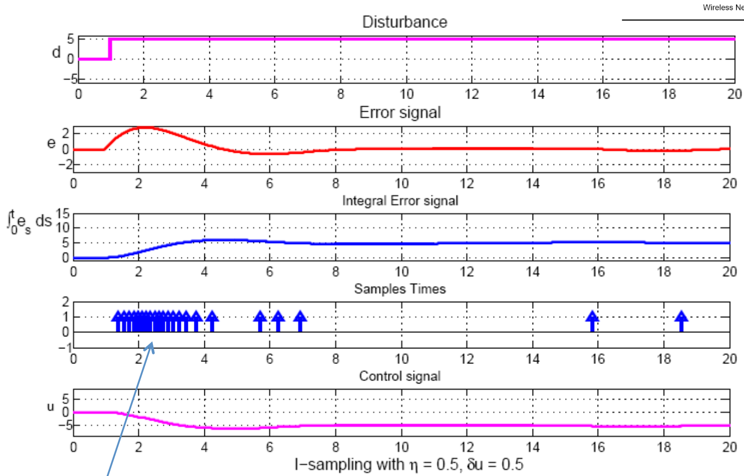


Control alphabet consists of three symbols, which are activated depending on the event

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