Tutorial Session: Event-triggered and Self-triggered Control

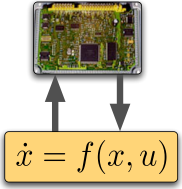
W.P.M.H. Heemels (TU/e)
K.H. Johansson (KTH)
Paulo Tabuada

Cyber-Physical Systems Laboratory
Department of Electrical Engineering
University of California at Los Angeles

Event-triggered and self-triggered control
Tutorial format

- Introduction to event-triggered and self-triggered control (P. Tabuada);
- Performance-based and output-based designs (W.P.M.H. Heemels);
- Stochastic approaches, wireless networking, and applications (K.H. Johansson).

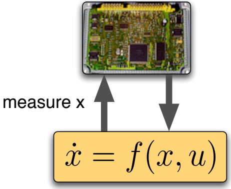
Feedback control loops are typically implemented on microprocessors.



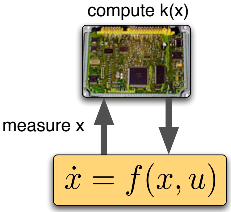
Introduction

Revisiting the standard assumptions and abstractions

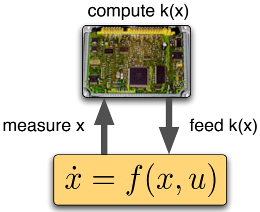
Feedback control loops are typically implemented on microprocessors.



Feedback control loops are typically implemented on microprocessors.



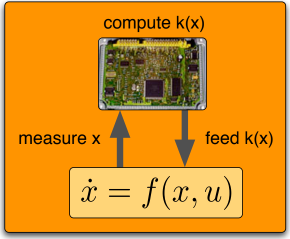
Feedback control loops are typically implemented on microprocessors.



Introduction

Revisiting the standard assumptions and abstractions

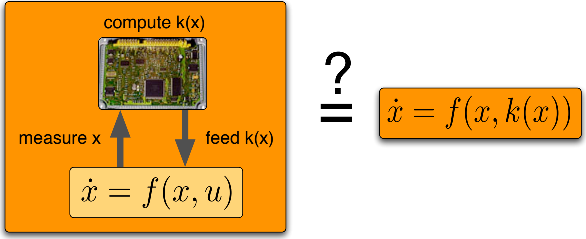
Feedback control loops are typically implemented on microprocessors.



$$\overset{?}{=} \quad \dot{x} = f(x, k(x))$$

However, most of existing control theory was developed by ignoring the implementation details.

Feedback control loops are typically implemented on microprocessors.

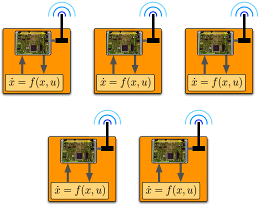


However, most of existing control theory was developed by ignoring the implementation details.

If the computation of $k(x)$ is sufficiently fast, if the sensors and actuators are sufficiently accurate, then the implementation (left box) will converge to the specification (right box).

Introduction

Revisiting the standard assumptions and abstractions

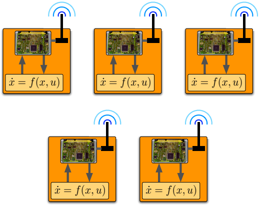


With the advent of networked embedded control systems, we can no longer rely on the assumption of dedicated hardware. Embedded systems are characterized by reduced computing and communicating capabilities. Available power is limited and has to be carefully managed.

We need a design theory integrating control, computation, and communication.

Introduction

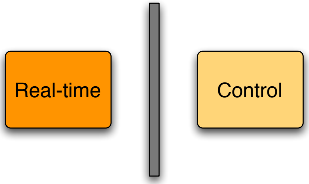
Revisiting the standard assumptions and abstractions



With the advent of networked embedded control systems, we can no longer rely on the assumption of dedicated hardware. Embedded systems are characterized by reduced computing and communicating capabilities. Available power is limited and has to be carefully managed.

We need a design theory integrating control, computation, and communication.

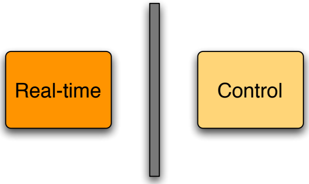
In this tutorial I will focus on the real-time requirements for control and their impact on computation and communication.



- The design and implementation of feedback control laws on microprocessors has traditionally been decoupled from real-time scheduling through a "separation of concerns" obtained by treating control tasks as periodic;

Introduction

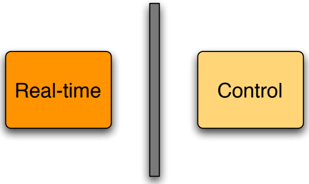
Interfacing Real-Time and Control



- The design and implementation of feedback control laws on microprocessors has traditionally been decoupled from real-time scheduling through a "separation of concerns" obtained by treating control tasks as periodic;
- Control engineers can design controllers while ignoring the implementations details;
- Software engineers can schedule tasks while ignoring their functionality;

Introduction

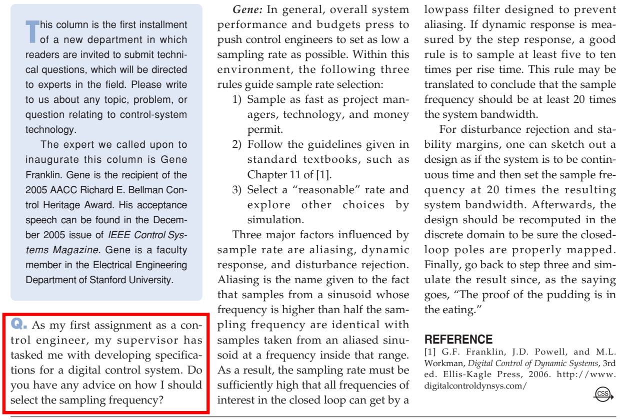
Interfacing Real-Time and Control



- The design and implementation of feedback control laws on microprocessors has traditionally been decoupled from real-time scheduling through a "separation of concerns" obtained by treating control tasks as periodic;
- Control engineers can design controllers while ignoring the implementations details;
- Software engineers can schedule tasks while ignoring their functionality;
- Although this "separation of concerns" simplifies the design process, it also results in inefficient usage of resources.

- From a purely theoretical perspective, executing control tasks in a periodic fashion seems unnatural and inconsiderate to the dynamics.
- Even if we accept the periodic paradigm, it is unsettling that we still do not understand how the sampling or execution period should be chosen!

IEEE Control Systems Magazine, 27(4), pp. 19, 2007.
Ask the experts column.



This column is the first installment of a new department in which readers are invited to submit technical questions, which will be directed to experts in the field. Please write to us about any topic, problem, or question relating to control-system technology.

The expert we called upon to inaugurate this column is Gene Franklin. Gene is the recipient of the 2005 AACC Richard E. Bellman Control Heritage Award. His acceptance speech can be found in the December 2005 issue of *IEEE Control Systems Magazine*. Gene is a faculty member in the Electrical Engineering Department of Stanford University.

Q. As my first assignment as a control engineer, my supervisor has tasked me with developing specifications for a digital control system. Do you have any advice on how I should select the sampling frequency?

Gene: In general, overall system performance and budgets press to push control engineers to set as low a sampling rate as possible. Within this environment, the following three rules guide sample rate selection:

1) Sample as fast as project managers, technology, and money permit.
2) Follow the guidelines given in standard textbooks, such as Chapter 11 of [1].
3) Select a "reasonable" rate and explore other choices by simulation.

Three major factors influenced by sample rate are aliasing, dynamic response, and disturbance rejection. Aliasing is the name given to the fact that samples from a sinusoid whose frequency is higher than half the sampling frequency are identical with samples taken from an aliased sinusoid at a frequency inside that range. As a result, the sampling rate must be sufficiently high that all frequencies of interest in the closed loop can get by a lowpass filter designed to prevent aliasing. If dynamic response is measured by the step response, a good rule is to sample at least five to ten times per rise time. This rule may be translated to conclude that the sample frequency should be at least 20 times the system bandwidth.

For disturbance rejection and stability margins, one can sketch out a design as if the system is to be continuous time and then set the sample frequency at 20 times the resulting system bandwidth. Afterwards, the design should be recomputed in the discrete domain to be sure the closed-loop poles are properly mapped. Finally, go back to step three and simulate the result since, as the saying goes, "The proof of the pudding is in the eating."

REFERENCE

[1] G.F. Franklin, J.D. Powell, and M.L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Ellis-Kagle Press, 2006. http://www.digitalcontroldynsys.com/

Introduction

Interfacing Real-Time and Control

IEEE Control Systems Magazine, 27(4), pp. 19, 2007.
Ask the experts column.

This column is the first installment of a new department in which readers are invited to submit technical questions, which will be directed to experts in the field. Please write to us about any topic, problem, or question relating to control-system technology.

The expert we called upon to inaugurate this column is Gene Franklin. Gene is the recipient of the 2005 AACC Richard E. Bellman Control Heritage Award. His acceptance speech can be found in the December 2005 issue of *IEEE Control Systems Magazine*. Gene is a faculty member in the Electrical Engineering Department of Stanford University.

Q. As my first assignment as a control engineer, my supervisor has tasked me with developing specifications for a digital control system. Do you have any advice on how I should select the sampling frequency?

Gene: In general, overall system performance and budgets press to push control engineers to set as low a sampling rate as possible. Within this environment, the following three rules guide sample rate selection:

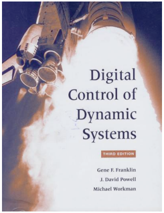
1) Sample as fast as project managers, technology, and money permit.
2) Follow the guidelines given in standard textbooks, such as Chapter 11 of [1].
3) Select a "reasonable" rate and explore other choices by simulation.

Three major factors influenced by sample rate are aliasing, dynamic response, and disturbance rejection. Aliasing is the name given to the fact that samples from a sinusoid whose frequency is higher than half the sampling frequency are identical with samples taken from an aliased sinusoid at a frequency inside that range. As a result, the sampling rate must be sufficiently high that all frequencies of interest in the closed loop can get by a lowpass filter designed to prevent aliasing. If dynamic response is measured by the step response, a good rule is to sample at least five to ten times per rise time. This rule may be translated to conclude that the sample frequency should be at least 20 times the system bandwidth.

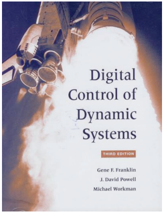
For disturbance rejection and stability margins, one can sketch out a design as if the system is to be continuous time and then set the sample frequency at 20 times the resulting system bandwidth. Afterwards, the design should be recomputed in the discrete domain to be sure the closed-loop poles are properly mapped. Finally, go back to step three and simulate the result since, as the saying goes, "The proof of the pudding is in the eating."

REFERENCE

[1] G.F. Franklin, J.D. Powell, and M.L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Ellis-Kagle Press, 2006. http://www.digitalcontroldynsys.com/



"... at least 20 times the closed-loop bandwidth..."



"... at least 20 times the closed-loop bandwidth..."

"In general, the best sampling which can be chosen for a digital control system is the slowest rate that meets all the performance requirements."

$$\frac{1}{30 f_c} < T < \frac{1}{5 f_c}$$

In this talk:

Abandon the periodic paradigm in favor of event-triggered and self-triggered control.

In this talk:

Abandon the periodic paradigm in favor of event-triggered and self-triggered control.

Why?

In this talk:

Abandon the periodic paradigm in favor of event-triggered and self-triggered control.

Why?

- In the context of sensor-actuator networks we would like to execute control tasks as rarely as possible in order to minimize energy consumption due to communication.
- Even if energy is not a concern, the less often control tasks are executed the more processor time is available for other (less) important tasks;
- Ideally, the scheduler should be able to dynamically adjust the quality of control (and implicitly the execution times, deadlines, ...) to respond to overloads and other transients.

In this talk:

Abandon the periodic paradigm in favor of event-triggered and self-triggered control.

Why?

- In the context of sensor-actuator networks we would like to execute control tasks as rarely as possible in order to minimize energy consumption due to communication.
- Even if energy is not a concern, the less often control tasks are executed the more processor time is available for other (less) important tasks;
- Ideally, the scheduler should be able to dynamically adjust the quality of control (and implicitly the execution times, deadlines, ...) to respond to overloads and other transients.

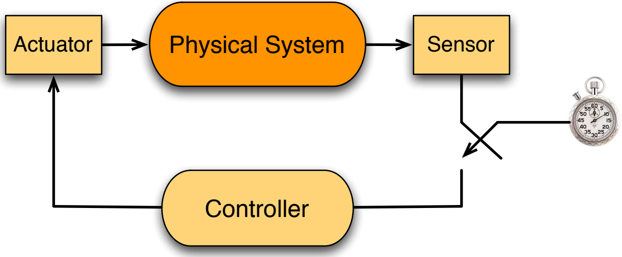
Event-triggered control had been advocated before. Two “recent” references that spurred our interest are:

- K.E. Arzen. A simple event based PID controller. 14th IFAC World Congress, 1999.
- K.J. Astrom and B.M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. 41st IEEE Conference on Decision and Control, 2002.

Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

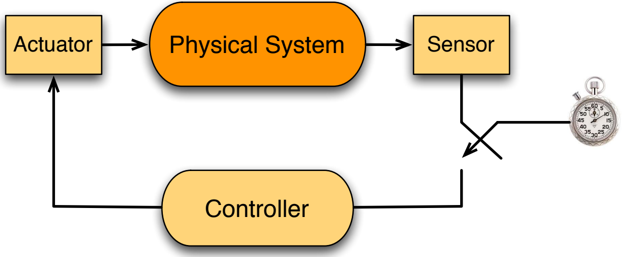
In time-triggered control the sensing, control, and actuation are driven by a clock.



Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

In time-triggered control the sensing, control, and actuation are driven by a clock.

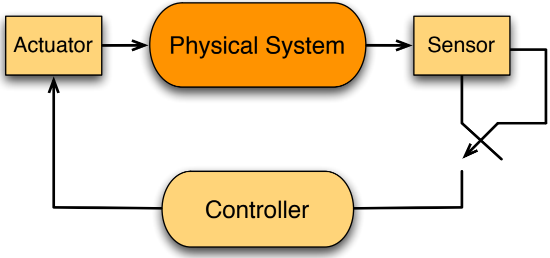


The periodic paradigm can be seen as *open-loop* sampling!

Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

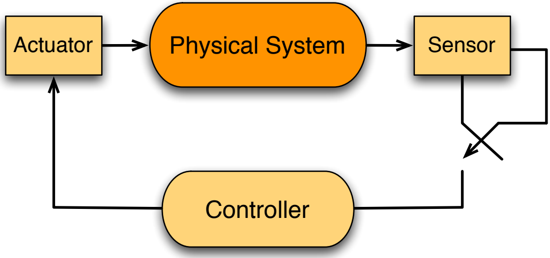
In event-triggered control the input is held constant, not periodically, but while performance is satisfactory.



Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

In event-triggered control the input is held constant, not periodically, but while performance is satisfactory.

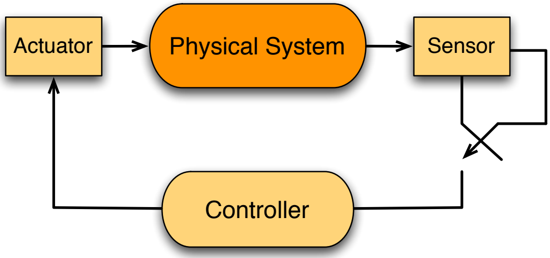


This can be regarded as introducing *feedback* in the sampling process!

Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

In event-triggered control the input is held constant, not periodically, but while performance is satisfactory.

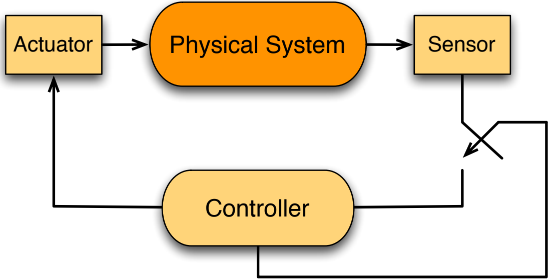


This can be regarded as introducing *feedback* in the sampling process!
Event-triggered control requires the constant monitoring of the state to determine current performance.

Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

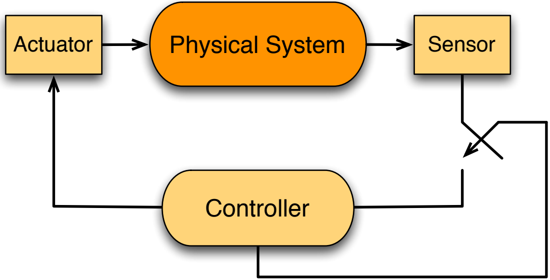
In self-triggered control, the current state is used not only to compute the input to the system, but also the next time the control law should be recomputed;



Periodic vs Event-triggered vs Self-triggered control

A new look at old ideas

In self-triggered control, the current state is used not only to compute the input to the system, but also the next time the control law should be recomputed;



Constant monitoring of the state is no longer needed although the loop is still closed based on current performance.

Event-triggered control

When to control?

Start with a linear control system:

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

and a linear feedback control law $u = Kx$ rendering the closed loop system asymptotically stable.

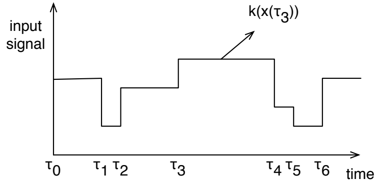
Start with a linear control system:

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

and a linear feedback control law $u = Kx$ rendering the closed loop system asymptotically stable.

For a given sequence of execution times $\tau_1, \tau_2, \tau_3, \ldots$ the control signal $u(t)$ and the error $e(t)$ are defined by:

$$u(t) = Kx(\tau_i) \qquad \text{for } \tau_i \leq t < \tau_{i+1}$$
$$e(t) = x(\tau_i) - x(t) \qquad \text{for } \tau_i \leq t < \tau_{i+1}$$



Start with a linear control system:

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

and a linear feedback control law $u = Kx$ rendering the closed loop system asymptotically stable.

Using the error $e(t) = x(\tau_i) - x(t)$ we can rewrite the closed loop dynamics as:

$$\dot{x}(t) \quad = \quad Ax(t) + BKx(\tau_i)$$

Event-triggered control

When to control?

Start with a linear control system:

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

and a linear feedback control law $u = Kx$ rendering the closed loop system asymptotically stable.

Using the error $e(t) = x(\tau_i) - x(t)$ we can rewrite the closed loop dynamics as:

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + BKx(\tau_i) \\
&= Ax(t) + BKx(t) - BKx(t) + BKx(\tau_i) \\
&= (A + BK)x(t) + BKe(t)
\end{aligned}$$

Event-triggered control

When to control?

Start with a linear control system:

$$\dot{x} = Ax + Bu, \qquad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

and a linear feedback control law $u = Kx$ rendering the closed loop system asymptotically stable.

Using the error $e(t) = x(\tau_i) - x(t)$ we can rewrite the closed loop dynamics as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + BKx(\tau_i) \\ &= Ax(t) + BKx(t) - BKx(t) + BKx(\tau_i) \\ &= (A + BK)x(t) + BKe(t) \end{aligned}$$

Since $A + BK$ is a stable matrix there exists a Lyapunov function V whose derivative along the closed loop dynamics satisfies:

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\|$$

Since $A + BK$ is a stable matrix there exists a Lyapunov function V whose derivative along the closed loop dynamics satisfies:

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\| \tag{1}$$

Assume now that we can enforce the inequality:

$$\|e\| \leq \sigma\|x\| \tag{2}$$

for some σ satisfying $-a + b\sigma < -a'$ with $a' > 0$. It would then follow from (1) and (2):

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\| \leq -a\|x\|^2 + b\sigma\|x\|^2 \leq -a'\|x\|^2$$

Event-triggered control

When to control?

Since $A + BK$ is a stable matrix there exists a Lyapunov function V whose derivative along the closed loop dynamics satisfies:

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\| \tag{1}$$

Assume now that we can enforce the inequality:

$$\|e\| \leq \sigma\|x\| \tag{2}$$

for some σ satisfying $-a + b\sigma < -a'$ with $a' > 0$. It would then follow from (1) and (2):

$$\dot{V} \leq -a\|x\|^2 + b\|x\|\|e\| \leq -a\|x\|^2 + b\sigma\|x\|^2 \leq -a'\|x\|^2$$

This suggests that $\|e\| = \sigma\|x\|$ could be used to decide when to execute the control task. By executing the control task at $t = \tau$ we obtain $e(t) = x(\tau) - x(t) = 0$ which implies (2).

We call $\|e\| = \sigma\|x\|$ an event triggering condition.

Event-triggered control

When to control?

Executing the control task when $\|e\| = \sigma\|x\|$ is satisfied guarantees stability and performance (as measured by a') but it also raises two questions:

1. The execution policy is not explicit in time. How do we know if the execution times will not have an accumulation point? How do we know if there is enough time to execute the control task?
2. How do we test the equality $\|e\| = \sigma\|x\|$?

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

Upon closing the loop at time τ_i we have $e(\tau_i) = 0$ and thus:

$$\frac{\|e(\tau_i)\|}{\|x(\tau_i)\|} = 0.$$

The loop will be closed again when:

$$\frac{\|e(t)\|}{\|x(t)\|} = \sigma.$$

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

The linear case
Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

$$\frac{d}{dt}\frac{\|e\|^2}{\|x\|^2} =$$

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

$$\frac{d}{dt}\frac{\|e\|^2}{\|x\|^2} = \frac{d}{dt}\frac{e^T e}{x^T x} = \frac{2e^T \dot{e} x^T x - 2x^T \dot{x} e^T e}{x^T x x^T x}$$

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i\in\mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

$$\begin{aligned} \frac{d}{dt}\frac{\|e\|^2}{\|x\|^2} &= \frac{d}{dt}\frac{e^Te}{x^Tx} = \frac{2e^T\dot{e}x^Tx - 2x^T\dot{x}e^Te}{x^Txx^Tx} \\ &= \frac{-2e^T((A+BK)x + BKe)x^Tx - 2x^T((A+BK)x + BKe)e^Te}{x^Txx^Tx} \end{aligned}$$

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

$$
\begin{aligned}
\frac{d}{dt}\frac{\|e\|^2}{\|x\|^2} &= \frac{d}{dt}\frac{e^Te}{x^Tx} = \frac{2e^T\dot{e}x^Tx - 2x^T\dot{x}e^Te}{x^Txx^Tx} \\
&= \frac{-2e^T((A+BK)x + BKe)x^Tx - 2x^T((A+BK)x + BKe)e^Te}{x^Txx^Tx} \\
&\leq 2\|A+BK\|\frac{\|e\|}{\|x\|} + 2(\|BK\| + \|A+BK\|)\frac{\|e\|^2}{\|x\|^2} + 2\|BK\|\frac{\|e\|^3}{\|x\|^3}
\end{aligned}
$$

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

How long does it take for $\|e\|/\|x\|$ to evolve from 0 to σ?

$$\begin{aligned}
\frac{d}{dt}\frac{\|e\|^2}{\|x\|^2} &= \frac{d}{dt}\frac{e^Te}{x^Tx} = \frac{2e^T\dot{e}x^Tx - 2x^T\dot{x}e^Te}{x^Txx^Tx} \\
&= \frac{-2e^T((A+BK)x + BKe)x^Tx - 2x^T((A+BK)x + BKe)e^Te}{x^Txx^Tx} \\
&\leq 2\|A+BK\|\frac{\|e\|}{\|x\|} + 2(\|BK\| + \|A+BK\|)\frac{\|e\|^2}{\|x\|^2} + 2\|BK\|\frac{\|e\|^3}{\|x\|^3}
\end{aligned}$$

The solutions of this differential inequality starting at 0 need at least τ^* units of time to reach σ^2.

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

- The minimum time τ^* can be used for periodic implementations.

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma \|x\|$.

- The minimum time τ^* can be used for periodic implementations.
- Stabilization tasks do not really need hard real-time. The control task can always be executed before $\|e\| = \sigma \|x\|$ and after.

The linear case

Inter-execution times

Theorem

Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

- The minimum time τ^* can be used for periodic implementations.
- Stabilization tasks do not really need hard real-time. The control task can always be executed before $\|e\| = \sigma\|x\|$ and after.
- This suggest that σ represents a trade-off between deadlines and quality of control as measured by the rate of decay of V (more on quality of control in the second lecture).

The linear case

Inter-execution times

Theorem

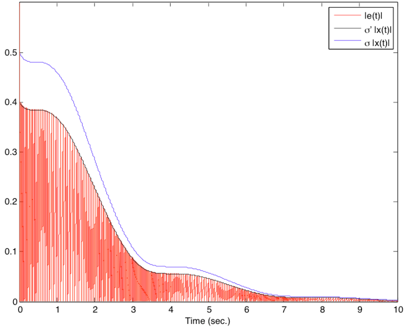
Let $u = Kx$ be a linear control law rendering the closed-loop system $\dot{x} = Ax + BKx$ asymptotically stable. For any $\sigma > 0$, there exists a lower bound $\tau^* \in \mathbb{R}^+$ for the inter-execution times $\{\tau_{i+1} - \tau_i\}_{i \in \mathbb{N}}$ implicitly defined by the event-triggering condition $\|e\| = \sigma\|x\|$.

- The minimum time τ^* can be used for periodic implementations.
- Stabilization tasks do not really need hard real-time. The control task can always be executed before $\|e\| = \sigma\|x\|$ and after.
- This suggest that σ represents a trade-off between deadlines and quality of control as measured by the rate of decay of V (more on quality of control in the second lecture).
- A similar result holds in the nonlinear case under suitable Lipschitz continuity assumptions.

The linear case

Example with non-zero execution time

If Δ is the worst case execution time of the control task then, given a desired σ defining the scheduling strategy, we can compute a $\sigma' < \sigma$ defining the new scheduling strategy $\|e\| = \sigma'\|x\|$ by taking Δ into account.



$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$u = x_1 - 4x_2 \qquad \Delta = 0.002s$$

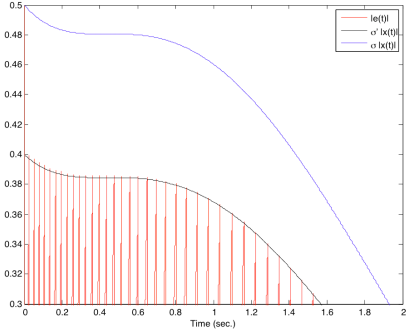
$$\sigma = 0.02 \qquad \sigma' = 0.0154$$

$$x(0) = (10, 20)$$

The linear case

Example with non-zero execution time

If Δ is the worst case execution time of the control task then, given a desired σ defining the scheduling strategy, we can compute a $\sigma' < \sigma$ defining the new scheduling strategy $\|e\| = \sigma'\|x\|$ by taking Δ into account.



$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$u = x_1 - 4x_2 \qquad \Delta = 0.002s$$

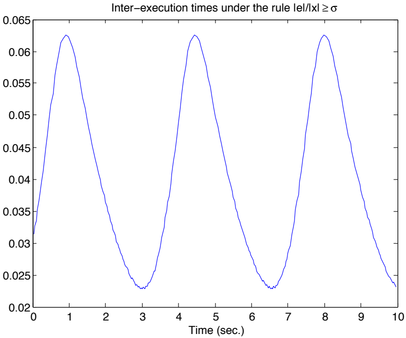
$$\sigma = 0.02 \qquad \sigma' = 0.0154$$

$$x(0) = (10, 20)$$

The linear case

Example with non-zero execution time

If Δ is the worst case execution time of the control task then, given a desired σ defining the scheduling strategy, we can compute a $\sigma' < \sigma$ defining the new scheduling strategy $\|e\| = \sigma'\|x\|$ by taking Δ into account.



$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$u = x_1 - 4x_2 \qquad \Delta = 0.002s$$

$$\sigma = 0.02 \qquad \sigma' = 0.0154$$

$$x(0) = (10, 20)$$

Event-triggered control

When to execute the control task

How do we implement the execution rule $\|e\| = \sigma\|x\|$?

Event-triggered control

When to execute the control task

How do we implement the execution rule $\|e\| = \sigma\|x\|$?

We can use dedicated hardware to test $\|e\| = \sigma\|x\|$ and generate an interrupt whenever this equality is satisfied.

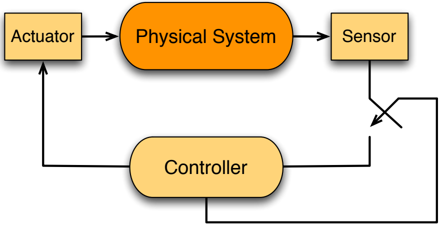
Event-triggered control

When to execute the control task

How do we implement the execution rule $\|e\| = \sigma\|x\|$?

We can use dedicated hardware to test $\|e\| = \sigma\|x\|$ and generate an interrupt whenever this equality is satisfied.

We can also emulate the event-triggered strategy by a self-triggered strategy:



- In the linear case we can integrate the dynamics to predict the next event;
- In the nonlinear case we have to be more creative.

Self-triggered control

The linear case

Given:

- A linear system $\dot{x} = Ax + Bu$;
- A stabilizing linear feedback control law $u = Kx$;
- A sampling rate $\mathfrak{t}$ required by the digital platform implementing the control law,

we construct the equivalent discrete-time model using constant inputs:

$$x(t + \mathfrak{t}) = A_{\mathfrak{t}} x(t) + B_{\mathfrak{t}} u(t), \quad u(t) = Kx(t).$$

Self-triggered control
The linear case

Given:

- A linear system $\dot{x} = Ax + Bu$;
- A stabilizing linear feedback control law $u = Kx$;
- A sampling rate $\mathfrak{t}$ required by the digital platform implementing the control law,

we construct the equivalent discrete-time model using constant inputs:

$$x(t + \mathfrak{t}) = A_{\mathfrak{t}} x(t) + B_{\mathfrak{t}} u(t), \quad u(t) = Kx(t).$$

We can now define the next execution time map $\tau : \mathbb{R}^n \to \mathbb{R}^+$ by the rule:

$$x(t) \mapsto t + k\mathfrak{t}$$

where k is the largest natural number for which:

$$V(x(t + \ell\mathfrak{t})) - V(x(t)) \leq -a\|x(t)\|, \quad \forall \ell \in \{1, 2, \ldots, k\}.$$

Self-triggered control

The linear case

Given:

- A linear system $\dot{x} = Ax + Bu$;
- A stabilizing linear feedback control law $u = Kx$;
- A sampling rate $\mathfrak{t}$ required by the digital platform implementing the control law,

we construct the equivalent discrete-time model using constant inputs:

$$x(t + \mathfrak{t}) = A_{\mathfrak{t}} x(t) + B_{\mathfrak{t}} u(t), \quad u(t) = Kx(t).$$

We can now define the next execution time map $\tau : \mathbb{R}^n \to \mathbb{R}^+$ by the rule:

$$x(t) \mapsto t + k\mathfrak{t}$$

where k is the largest natural number for which:

$$V(x(t + \ell\mathfrak{t})) - V(x(t)) \leq -a\|x(t)\|, \quad \forall \ell \in \{1, 2, \ldots, k\}.$$

Other variations possible: rather than using $u = Kx$, find the best u (more on this in the second lecture).

How about nonlinear systems?

[1] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

[1] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

Self-triggered control

The nonlinear case

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

Consider the control of the angular momentum for a rigid spacecraft with 2 inputs and a stabilizing control law[1] :

$$\begin{aligned} \dot{x}_1 &= u_1, \qquad u_1 = -x_1 x_2 - 2x_2 x_3 - x_1 - x_3 \\ \dot{x}_2 &= u_2, \qquad u_2 = 2x_1 x_2 x_3 + 3x_3^2 - x_2 \\ \dot{x}_3 &= x_1 x_2 \end{aligned}$$

[1] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

Self-triggered control

The nonlinear case

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

Consider the control of the angular momentum for a rigid spacecraft with 2 inputs and a stabilizing control law[2] :

$$\begin{aligned}
\dot{x}_1 &= -x_1 x_2 - 2x_2 x_3 - x_1 - x_3 \\
\dot{x}_2 &= 2x_1 x_2 x_3 + 3x_3^2 - x_2 \\
\dot{x}_3 &= x_1 x_2
\end{aligned}$$

[2] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

Self-triggered control

The nonlinear case

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

Consider the control of the angular momentum for a rigid spacecraft with 2 inputs and a stabilizing control law[3] :

$$
\begin{aligned}
\dot{x}_1 &= -x_1 x_2 z - 2x_2 x_3 z - x_1 z^2 - x_3 z^2 \\
\dot{x}_2 &= 2x_1 x_2 x_3 + 3x_3^2 z - x_2 z^2 \\
\dot{x}_3 &= x_1 x_2 z
\end{aligned}
$$

[3] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

Self-triggered control

The nonlinear case

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

Consider the control of the angular momentum for a rigid spacecraft with 2 inputs and a stabilizing control law[4] :

$$
\begin{aligned}
\dot{x}_1 &= -x_1 x_2 z - 2x_2 x_3 z - x_1 z^2 - x_3 z^2 \\
\dot{x}_2 &= 2x_1 x_2 x_3 + 3x_3^2 z - x_2 z^2 \\
\dot{x}_3 &= x_1 x_2 z \\
\dot{z} &= 0
\end{aligned}
$$

[4] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

How about nonlinear systems? Given:

- A polynomial control system $\dot{x} = f(x, u)$, $x \in \mathbb{R}^n$;
- A stabilizing polynomial feedback control law $u = k(x)$,

we can embed the closed-loop system $\dot{x} = f(x, k(x)) = f(x)$ into an homogeneous (all the monomials have the same degree) closed-loop system $\dot{x}' = f'(x')$, $x' \in \mathbb{R}^{n+1}$.

Consider the control of the angular momentum for a rigid spacecraft with 2 inputs and a stabilizing control law[4] :

$$
\begin{aligned}
\dot{x}_1 &= -x_1 x_2 z - 2x_2 x_3 z - x_1 z^2 - x_3 z^2 \\
\dot{x}_2 &= 2x_1 x_2 x_3 + 3x_3^2 z - x_2 z^2 \\
\dot{x}_3 &= x_1 x_2 z \\
\dot{z} &= 0
\end{aligned}
$$

Solutions of $\dot{x} = f(x)$ and $\dot{x}' = f'(x')$ are then same whenever $z(0) = 1$.

[4] C.I. Byrnes and A. Isidori. New results and examples in nonlinear feedback stabilization. Systems & Control Letters, 12(4):437-442, 1989.

Self-triggered control

The nonlinear case

Theorem (Scaling law)

Let $\dot{x} = f(x, u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

$$\tau\big(\lambda x(t)\big) = \frac{1}{\lambda^d}\tau\big(x(t)\big)$$

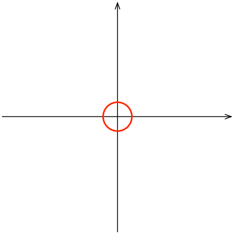
Self-triggered control

The nonlinear case

Theorem (Scaling law)

Let $\dot{x} = f(x, u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

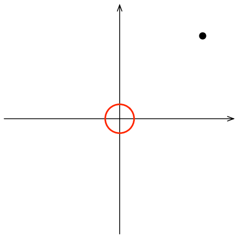
$$\tau\big(\lambda x(t)\big) = \frac{1}{\lambda^d}\tau\big(x(t)\big)$$



Theorem (Scaling law)

Let $\dot{x} = f(x,u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

$$\tau\big(\lambda x(t)\big) = \frac{1}{\lambda^d}\tau\big(x(t)\big)$$



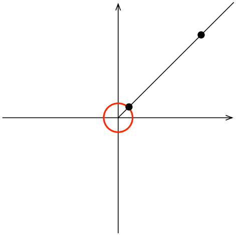
Self-triggered control

The nonlinear case

Theorem (Scaling law)

Let $\dot{x} = f(x, u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

$$\tau(\lambda x(t)) = \frac{1}{\lambda^d}\tau(x(t))$$



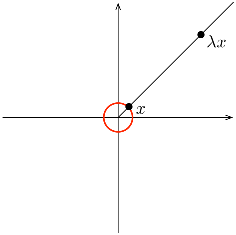
Self-triggered control

The nonlinear case

Theorem (Scaling law)

Let $\dot{x} = f(x, u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

$$\tau\big(\lambda x(t)\big) = \frac{1}{\lambda^d}\tau\big(x(t)\big)$$



Theorem (Scaling law)

Let $\dot{x} = f(x, u)$ be a control system for which a feedback control law $u = k(x)$, rendering the closed loop system homogeneous of order d, has been designed. The inter-execution times implicitly defined by the execution rule $\|e\| = \sigma\|x\|$ scale according to:

$$\tau\big(\lambda x(t)\big) = \frac{1}{\lambda^d}\tau\big(x(t)\big)$$

As the state approaches the origin, the inter-execution times $\tau(x(t))$ become larger.

Non-periodic executions are more efficient!

Consider again the control of the angular momentum for a rigid spacecraft with 2 inputs:

$$\begin{aligned} \dot{x}_1 &= -x_1 x_2 - 2x_2 x_3 - x_1 - x_3 \\ \dot{x}_2 &= 2x_1 x_2 x_3 + 3x_3^2 - x_2 \\ \dot{x}_3 &= x_1 x_2. \end{aligned}$$

Stability can be checked by resorting to:

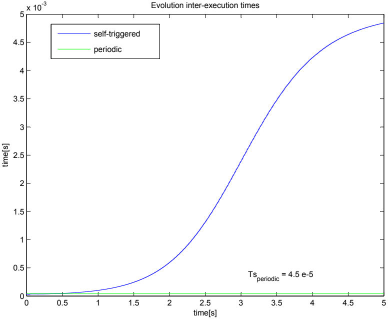
$$\begin{aligned} V &= \frac{1}{2}(x_1 + x_3)^2 + \frac{1}{2}(x_2 - x_3^2)^2 + x_3^2 \\ \frac{\partial V}{\partial x} f(x, k(x+e)) &\leq -91485\|x\|^2 + 147260\|x\|\|e\|. \end{aligned}$$

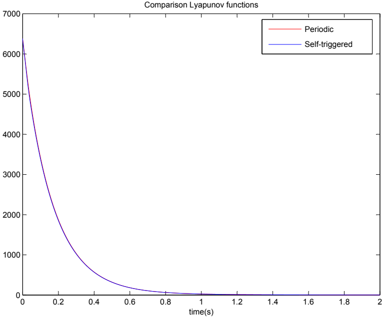
Self-triggered control

Polynomial example

Using $\sigma = 0.01$ we obtain $\tau^* = 5.2ms$ for a ball of radius 15 and:

$$\tau(x) = \frac{1}{1 + \|x\|^2}\tau^*$$





Self-triggered control

Polynomial example II (using a more general notion of homogeneity)

Consider the control of a jet engine compressor:

$$\begin{aligned}
\dot{x}_1 &= -x_2 - \frac{3}{2}x_1^2 - \frac{1}{2}x_1^3 \\
\dot{x}_2 &= \frac{1}{\beta^2}(x_1 - u) \\
u &= x_1 - \frac{\beta^2}{2}(x_1^2 + 1)(y + x_1^2 y + x_1 y^2) + 2\beta^2 x_1 \\
y &= 2\frac{x_1^2 + x_2}{x_1^2 + 1}
\end{aligned}$$

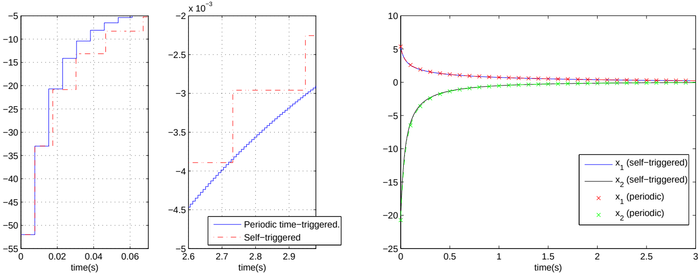
New deadline is a function of the current state of the physical system:

$$\tau(x_1, y) = \frac{29x_1 + x_1^2 + y^2}{5.36x_1^2\sqrt{x_1^2 + y^2} + x_1^2 + y^2} \cdot \tau^*.$$

Self-triggered control

Polynomial example II (using a more general notion of homogeneity)

Consider the control of a jet engine compressor:



New deadline is a function of the current state:

$$\tau(x_1, y) = \frac{29x_1 + x_1^2 + y^2}{5.36x_1^2\sqrt{x_1^2 + y^2} + x_1^2 + y^2} \cdot \tau^*.$$

- We use a Lyapunov function as a measure of performance. We would rather use a cost (more on this in the second lecture);

- We use a Lyapunov function as a measure of performance. We would rather use a cost (more on this in the second lecture);
- We assumed we have access to the state. Most often we cannot measure the full state (more on this in the second lecture);

Event-triggered and Self-triggered control

Important questions

- We use a Lyapunov function as a measure of performance. We would rather use a cost (more on this in the second lecture);
- We assumed we have access to the state. Most often we cannot measure the full state (more on this in the second lecture);
- Fully exploiting the potential of event-triggered and self-triggered control requires addressing scheduling problems for:
  - Single-processor systems;
  - Wired networks (e.g. CAN);
  - Wireless networks (more on this in the third lecture).

For papers and more information:

`http://www.cyphylab.ee.ucla.edu/`
`http://www.ee.ucla.edu/~tabuada`