Time and controls
(When to exercise control?)

Paulo Tabuada

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

Does control need real-time?

- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

Does control need real-time?

- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

But ...

Does control need real-time?

- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

But ... does control need real-time?

Does control need real-time?

- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

But ... does control need real-time?

- I believe so.

Does control need real-time?

- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

But ... does control need real-time?

- I believe so. However, we know very little about the role of *time* in control although there are far too many misconceptions.

Does control need real-time?

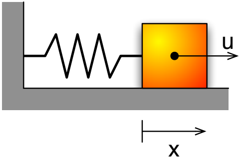
- Cyber-Physical Systems present new possibilities but also new control challenges.
- Chief among the challenges are real-time guarantees.

But ... does control need real-time?

- I believe so. However, we know very little about the role of *time* in control although there are far too many misconceptions.

Let me argue as we sip different cocktails of time and control.

We start with a textbook example to provide a concrete context for our cocktails.



$$
\begin{aligned}
\dot{x} &= v \\
\dot{v} &= -\frac{k}{m}x + \frac{1}{m}u
\end{aligned}
$$

The objective is to design a control law $u = Kx$ to stabilize the mass at $x = 0$. For simplicity we take $m = 1$ and $k = 10$.

Several alternatives are possible:

1. Periodic control (classical approach, first part of my talk);
2. Event-triggered, self-triggered control and beyond (second part of my talk).

Classical approach:

1. Design a controller ($u = -4v$) resulting in an asymptotically stable[1] closed-loop system assuming ideal sensors, ideal actuators, and instantaneous infinite precision computation.

[1] With eigenvalues at $\lambda = -2 \pm i\sqrt{6}$.

Cocktails of time and control

The first cocktail

Classical approach:

1. Design a controller ($u = -4v$) resulting in an asymptotically stable[1] closed-loop system assuming ideal sensors, ideal actuators, and instantaneous infinite precision computation.
2. Implement this controller in a periodic sample-and-hold manner.

[1] With eigenvalues at $\lambda = -2 \pm i\sqrt{6}$.

Classical approach:

1. Design a controller ($u = -4v$) resulting in an asymptotically stable[1] closed-loop system assuming ideal sensors, ideal actuators, and instantaneous infinite precision computation.
2. Implement this controller in a periodic sample-and-hold manner.

It is known that under a sufficiently fast periodic sample-and-hold implementation, the closed-loop system will retain stability.

[1] With eigenvalues at $\lambda = -2 \pm i\sqrt{6}$.

Classical approach:

1. Design a controller ($u = -4v$) resulting in an asymptotically stable[1] closed-loop system assuming ideal sensors, ideal actuators, and instantaneous infinite precision computation.
2. Implement this controller in a periodic sample-and-hold manner.

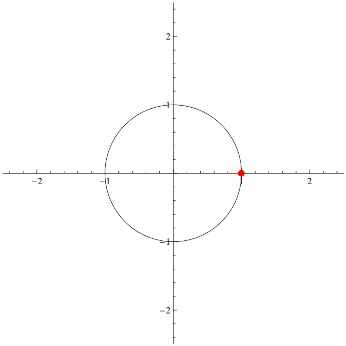
It is known that under a sufficiently fast periodic sample-and-hold implementation, the closed-loop system will retain stability.

Perhaps due to this result, it is widely believed that control requires short periods (high sampling rates) and that stability is lost if deadlines are missed or periods are increased.

[1] With eigenvalues at $\lambda = -2 \pm i\sqrt{6}$.

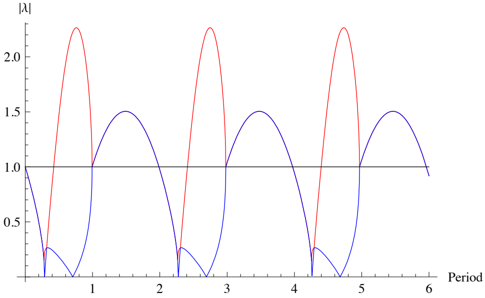
For a periodic implementation asymptotic stability is equivalent to the closed-loop eigenvalues being strictly inside the unit disk.

Let us plot the location of the eigenvalues as a function of the chosen period.



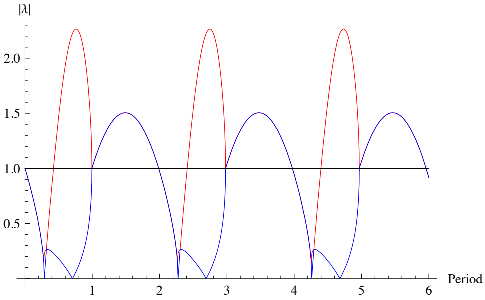
For a periodic implementation asymptotic stability is equivalent to the closed-loop eigenvalues being strictly inside the unit disk.

Let us plot the location of the eigenvalues as a function of the chosen period.



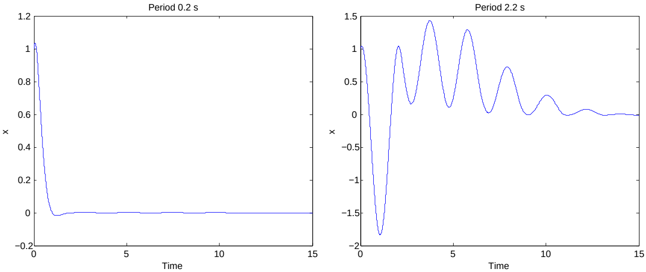
For a periodic implementation asymptotic stability is equivalent to the closed-loop eigenvalues being strictly inside the unit disk.

Let us plot the location of the eigenvalues as a function of the chosen period.

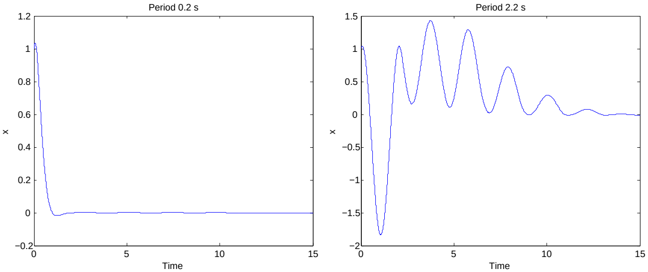


Stability is achieved with small periods as well as with (arbitrarily) large periods!

Stability, however, is not the full story.

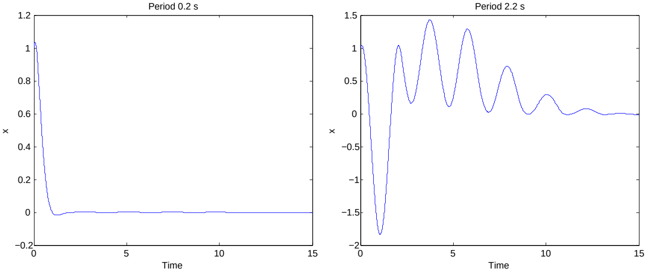


Stability, however, is not the full story.



Is performance monotonically increasing with decreasing periods?

Stability, however, is not the full story.

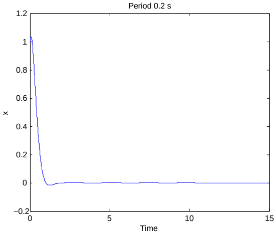


Is performance monotonically increasing with decreasing periods?

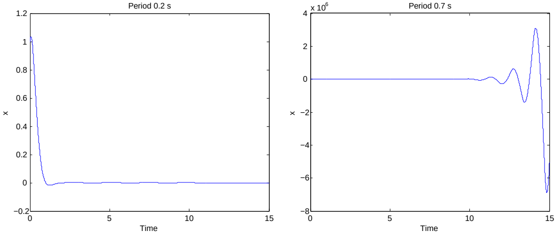
No! Shorter periods magnify the undesired effects of sensor noise and quantization[2].

[2] B. Bamieh. *Intersample and finite wordlength effects in sampled-data problems*. IEEE Transactions on Automatic Control, 48(4), 639–643, 2003.

Periods and performance, different cocktails for different tastes.



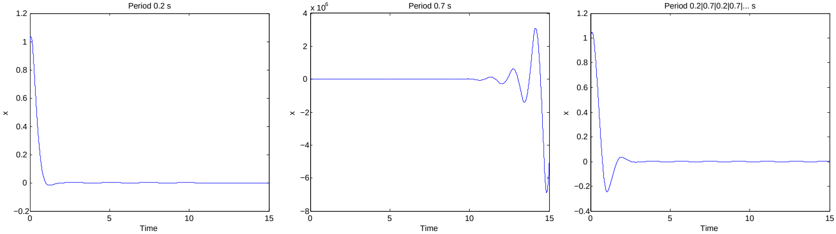
Periods and performance, different cocktails for different tastes.



Cocktails of time and control

The third cocktail: mixing periods

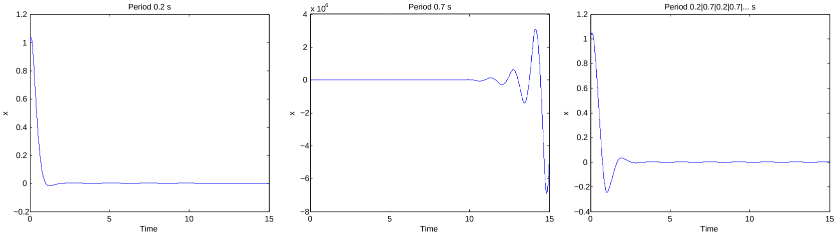
Periods and performance, different cocktails for different tastes.



Cocktails of time and control

The third cocktail: mixing periods

Periods and performance, different cocktails for different tastes.

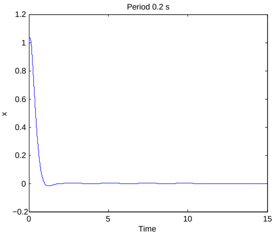


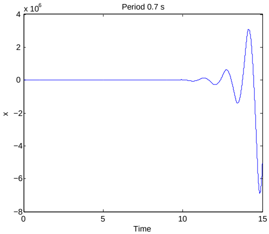
$$J(x(0), v(0)) = \int_0^\infty x^2(\tau) + v^2(\tau) + u^2(\tau) d\tau$$

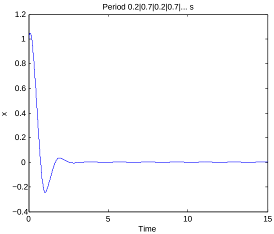
Cocktails of time and control

The third cocktail: mixing periods

Periods and performance, different cocktails for different tastes.





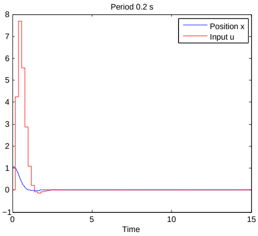


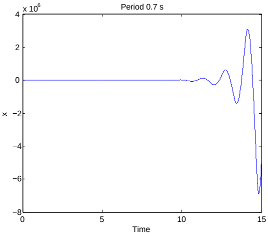
$J(1,1) \approx 25.4$

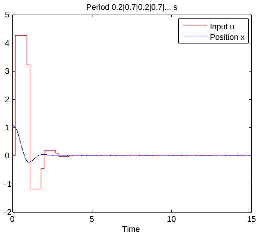
$J(1,1) \approx 18.5$

$$J(x(0), v(0)) = \int_0^\infty x^2(\tau) + v^2(\tau) + u^2(\tau) d\tau$$

Periods and performance, different cocktails for different tastes.





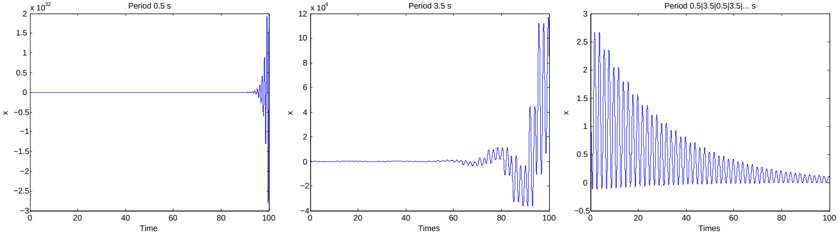


$J(1,1) \approx 25.4$

$J(1,1) \approx 18.5$

$$J(x(0), v(0)) = \int_0^\infty x^2(\tau) + v^2(\tau) + u^2(\tau) d\tau$$

We can even stabilize this system using only periods for which the system is unstable!



Cocktails of time and control

Reflections of an inebriated man

- The relation between time, stability, and performance is quite complex and not well understood[3];

[3] Beware of good looking (convex) Quality of Control or utility functions.

Cocktails of time and control

Reflections of an inebriated man

- The relation between time, stability, and performance is quite complex and not well understood[3];
- The periodic paradigm only offers a narrow view of the role of time in control. In particular, it fails to explain why we obtain better performance when using unstable periods;

3 Beware of good looking (convex) Quality of Control or utility functions.

Cocktails of time and control

- The relation between time, stability, and performance is quite complex and not well understood[3];
- The periodic paradigm only offers a narrow view of the role of time in control. In particular, it fails to explain why we obtain better performance when using unstable periods;
- The periodic paradigm results in a conservative usage of resources since the period is chosen for worst case operating conditions that rarely happen;

[3] Beware of good looking (convex) Quality of Control or utility functions.

Cocktails of time and control

Reflections of an inebriated man

- The relation between time, stability, and performance is quite complex and not well understood[3];
- The periodic paradigm only offers a narrow view of the role of time in control. In particular, it fails to explain why we obtain better performance when using unstable periods;
- The periodic paradigm results in a conservative usage of resources since the period is chosen for worst case operating conditions that rarely happen;
- The periodic paradigm can be seen as open-loop sampling;

[3] Beware of good looking (convex) Quality of Control or utility functions.

Cocktails of time and control

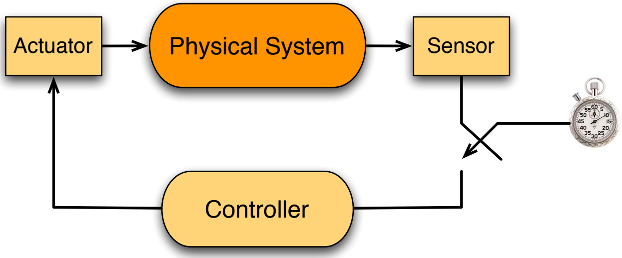
Reflections of an inebriated man

- The relation between time, stability, and performance is quite complex and not well understood[3];
- The periodic paradigm only offers a narrow view of the role of time in control. In particular, it fails to explain why we obtain better performance when using unstable periods;
- The periodic paradigm results in a conservative usage of resources since the period is chosen for worst case operating conditions that rarely happen;
- The periodic paradigm can be seen as open-loop sampling;
- Recently, event-triggered and self-triggered control have been proposed as a way of introducing *feedback* in the sampling process.

---

[3] Beware of good looking (convex) Quality of Control or utility functions.

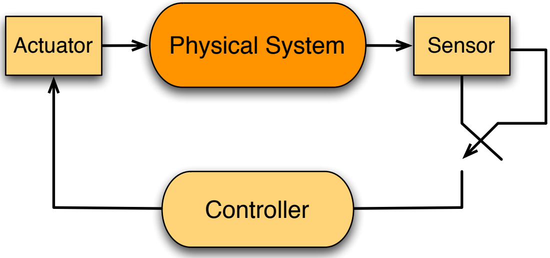
In time-triggered control the sensing, control, and actuation are driven by a clock.



Event-triggered and self-triggered control

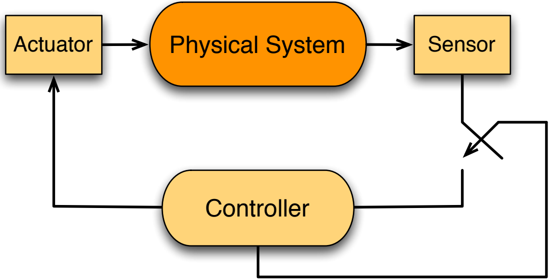
A new look at old ideas

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



This requires the constant monitoring of the state to determine the current performance.

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 the current performance.

As an example we consider the control of a jet engine compressor:

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

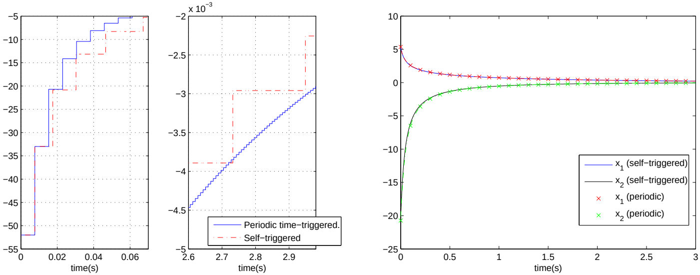
As an example we consider the control of a jet engine compressor:

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

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

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

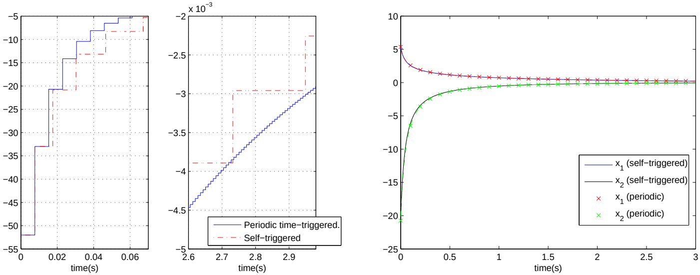
As an example we consider the control of a jet engine compressor:



New deadline is a function of the current state:

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

As an example we consider the control of a jet engine compressor:



New deadline is a function of the current state:

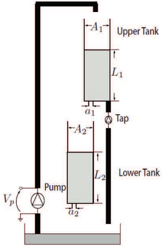
$$\tau(x_1, y) = \frac{29x_1 + x_1^2 + y^2}{5.36x_1^2\sqrt{x_1^2 + y^2} + x_1^2 + y^2} \cdot \tau^*.$$

It works in theory... How about in practice?

Event-triggered and self-triggered control

Welcome to the real world!

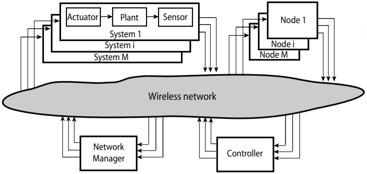
In collaboration with Kalle Johansson at KTH we tested these ideas in a real deployment.

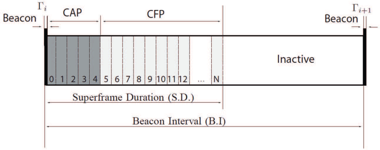


- Two independent water tanks controlled over the same wireless network.
- 1 sensing, 1 controller, and 1 actuator node per water tank.
- Additional temperature and humidity sensors on the wall to generate low priority traffic.

Event-triggered and self-triggered control
Welcome to the real world!

In collaboration with Kalle Johansson at KTH we tested these ideas in a real deployment.



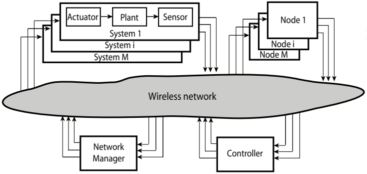


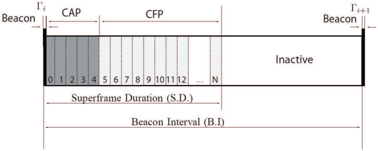
- Slight modification of the IEEE 802.15.4 MAC protocol (used in the proposed WirelessHART);
- The IEEE 802.15.4 MAC protocol offers a Contention Access Period (CAP) using CSMA/CA and a Contention Free Period (CFP) using TDMA;
- Telos motes with 250kbps 2.4 GHz Chipcon CC2420 IEEE 802.15.4 compliant radio.

Event-triggered and self-triggered control

Welcome to the real world!

In collaboration with Kalle Johansson at KTH we tested these ideas in a real deployment.



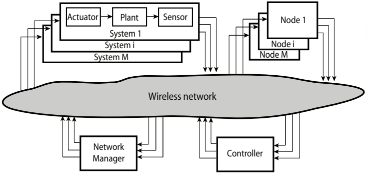


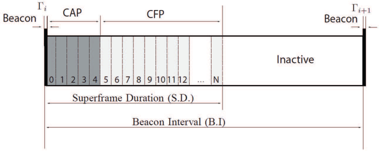
Scheme	Integral of Absolute Error	#updates in 220 s.	Battery life (days)
Event-triggered	88.26	49	879.84
Self-triggered	101.65	34	1010.18
Hybrid	85.06	35	910.14
Periodic	88.50	347	636.81

Event-triggered and self-triggered control

Welcome to the real world!

In collaboration with Kalle Johansson at KTH we tested these ideas in a real deployment.





Scheme	Integral of Absolute Error	#updates in 220 s.	Battery life (days)
Event-triggered	88.26	49	879.84
Self-triggered	101.65	34	1010.18
Hybrid	85.06	35	910.14
Periodic	88.50	347	636.81

Beyond event-triggered and self-triggered control

- Most event-triggered and self-triggered control designs are still based on a controller designed in oblivion of the implementation details;

Beyond event-triggered and self-triggered control

- Most event-triggered and self-triggered control designs are still based on a controller designed in oblivion of the implementation details;
- We can go one step further:
  - Minimum attention control: select the input that postpones as much as possible the re-computation of the input, subject to a desired level of control performance;

Beyond event-triggered and self-triggered control

- Most event-triggered and self-triggered control designs are still based on a controller designed in oblivion of the implementation details;
- We can go one step further:
  - Minimum attention control: select the input that postpones as much as possible the re-computation of the input, subject to a desired level of control performance;
  - Anytime attention control: based on a deadline provided by the real-time scheduler, compute an input that does not need to be changed before a given deadline while maximizing control performance;

Beyond event-triggered and self-triggered control

- Most event-triggered and self-triggered control designs are still based on a controller designed in oblivion of the implementation details;
- We can go one step further:
  - Minimum attention control: select the input that postpones as much as possible the re-computation of the input, subject to a desired level of control performance;
  - Anytime attention control: based on a deadline provided by the real-time scheduler, compute an input that does not need to be changed before a given deadline while maximizing control performance;
  - Asynchronous control: design an observer/controller that works even when receiving out-of-order sensor measurements.

Final thoughts

- The periodic paradigm has served us well, but Cyber-Physical Systems claim for more versatile paradigms;

Final thoughts

- The periodic paradigm has served us well, but Cyber-Physical Systems claim for more versatile paradigms;
- Event-triggered, self-triggered, minimum attention, anytime attention, and asynchronous control are attempts to go beyond periodicity;

Final thoughts

- The periodic paradigm has served us well, but Cyber-Physical Systems claim for more versatile paradigms;
- Event-triggered, self-triggered, minimum attention, anytime attention, and asynchronous control are attempts to go beyond periodicity;
- Fully exploiting the promise of these new paradigms requires a radically new clockwork:
  - How to provide real-time guarantees when events are only known at execution time?
  - How to dynamically trade computation and communication resources for performance?
  - How to mix time-triggered with event-triggered controllers?
  - ...

Final thoughts

- The periodic paradigm has served us well, but Cyber-Physical Systems claim for more versatile paradigms;
- Event-triggered, self-triggered, minimum attention, anytime attention, and asynchronous control are attempts to go beyond periodicity;
- Fully exploiting the promise of these new paradigms requires a radically new clockwork:
    - How to provide real-time guarantees when events are only known at execution time?
    - How to dynamically trade computation and communication resources for performance?
    - How to mix time-triggered with event-triggered controllers?
    - ...

Does control need real-time?

Final thoughts

- The periodic paradigm has served us well, but Cyber-Physical Systems claim for more versatile paradigms;
- Event-triggered, self-triggered, minimum attention, anytime attention, and asynchronous control are attempts to go beyond periodicity;
- Fully exploiting the promise of these new paradigms requires a radically new clockwork:
  - How to provide real-time guarantees when events are only known at execution time?
  - How to dynamically trade computation and communication resources for performance?
  - How to mix time-triggered with event-triggered controllers?
  - ...

Does control need real-time?

Yes, but not the *old clockwork* based on periods and deadlines!

For papers and more information:
`http://www.cyphylab.ee.ucla.edu`