Why Quantified Insight in System Design is Required.

by *Gerrit Muller* Buskerud University College
e-mail: gaudisite@gmail.com
www.gaudisite.nl

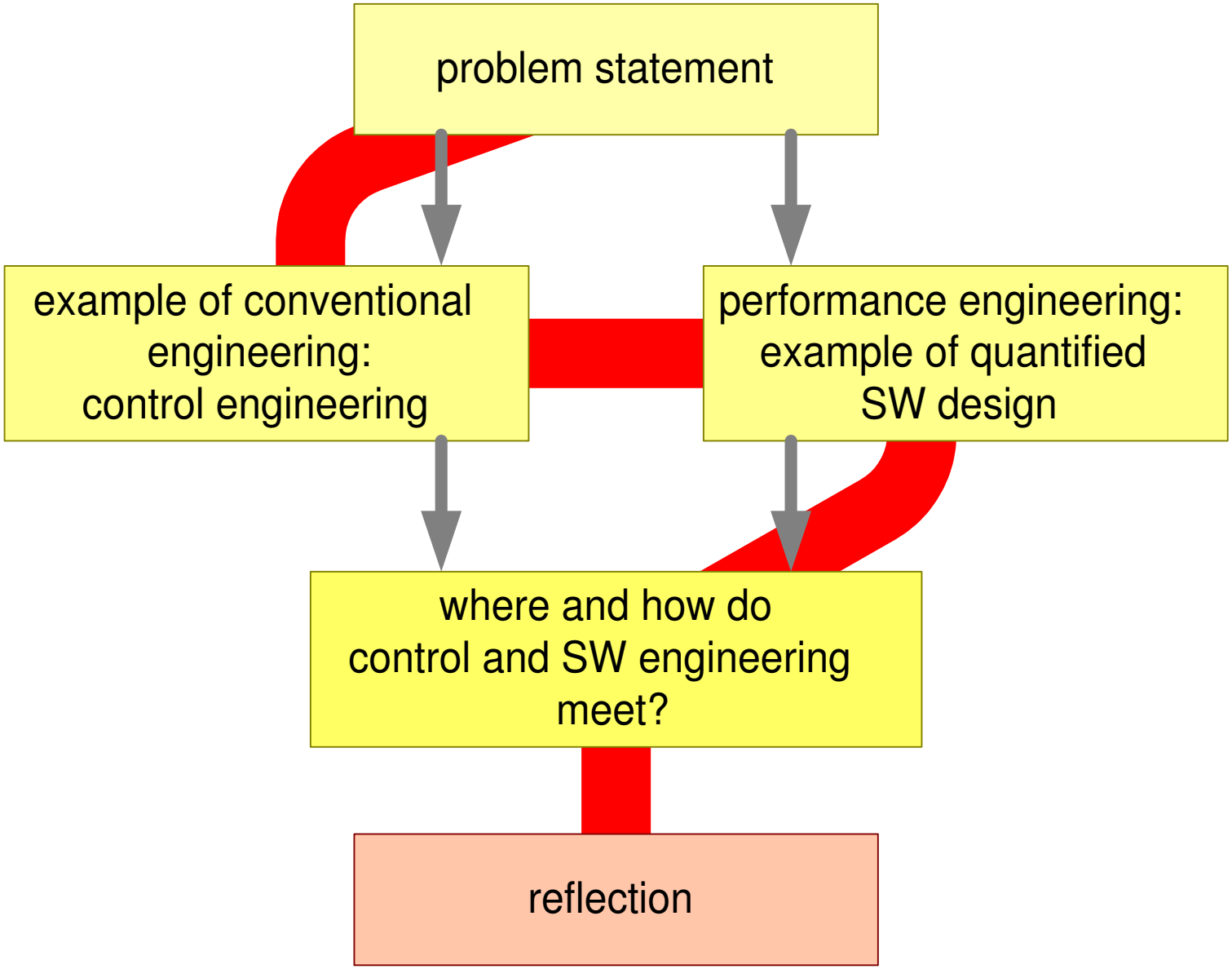
Abstract

Software education is mostly function-oriented. Techniques and formalisms are focused on system *behavior*. Software architects often don't have a quantified insight in problem domain or choosen solutions, although computers work internally with bits and bytes. This is a problem for IT systems in general, but is more so for embedded systems. Embedded systems interact with the physical world, which can be modeled quantitatively: energy consumption, speed, force, et cetera. This presentation addresses quantification of system and software design, illustrated by case examples.

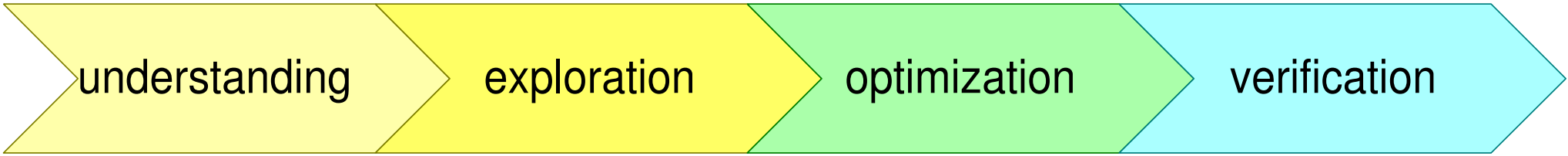
July 4, 2016
status: preliminary draft
version: 0.2

LANDELIJK architectuur congres ARCHITECTUUR VOOR DE WENDBARE ORGANISATIE

Figure Of Contents™



Purpose of Quantification



Challenge

Ask a SW-architect to *quantify* the product under construction.

What happens?

?

Challenge; Answer to the Question

Ask a SW-architect to *quantify* the product under construction.

What happens?

The *project* is quantified, rather than the *system* of interest

man-years
lines-of-code
problem reports

code-complexity
fault density
release schedule

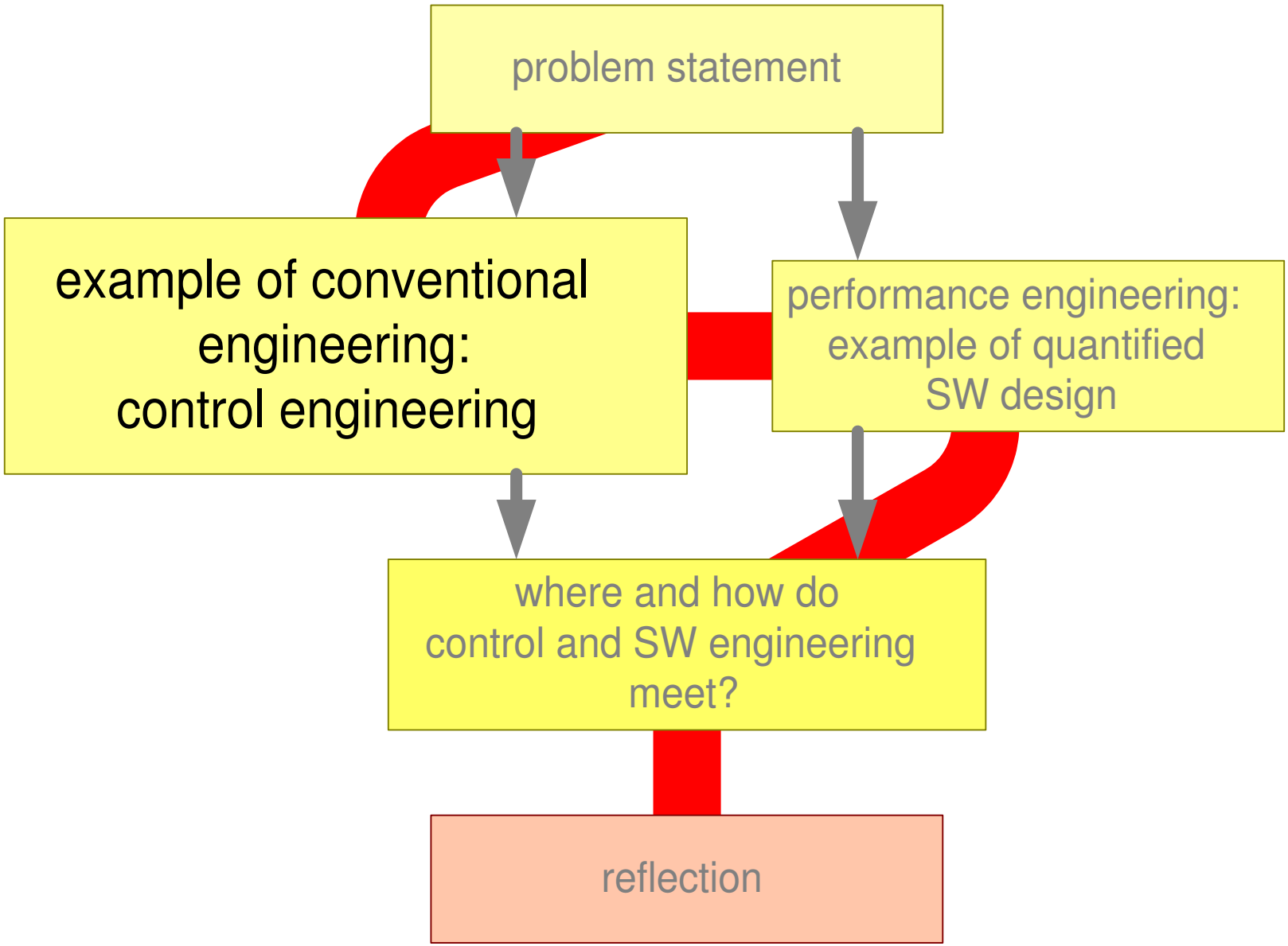
Problem Statement

The SW engineering discipline today is *process* oriented, quantities are process metrics.

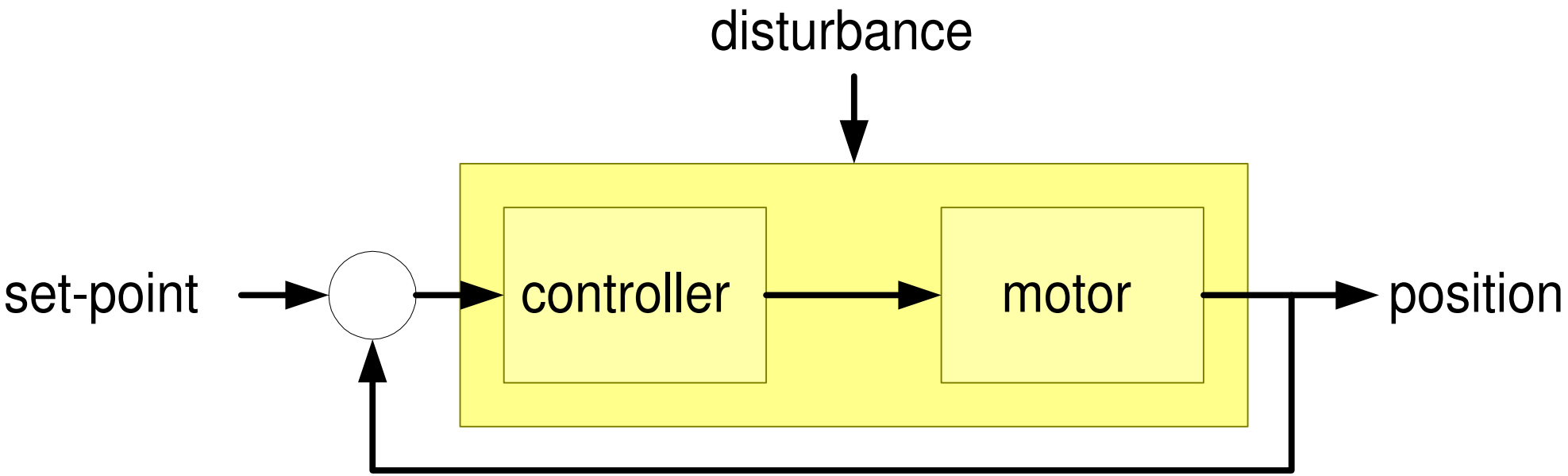
The System Of Interest (SOI) is designed from *behavioral* point of view.

Conventional Engineering disciplines design the SOI with *quantitative* techniques.

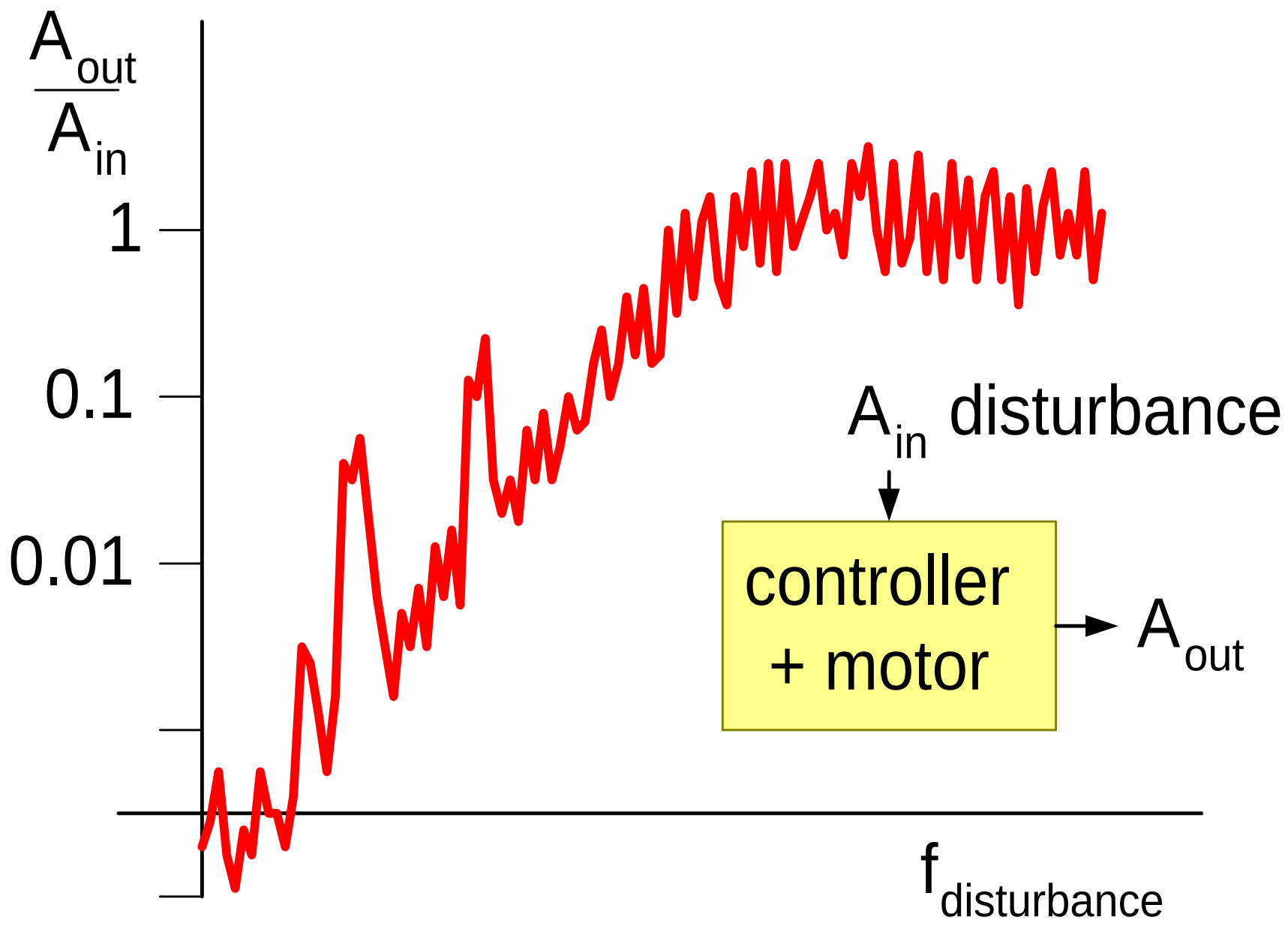
Qualities of SW intensive systems, such as performance, are *emerging* i.s.o. *predictable* properties


problem statement
example of conventional engineering: control engineering
performance engineering: example of quantified SW design
where and how do control and SW engineering meet?
reflection

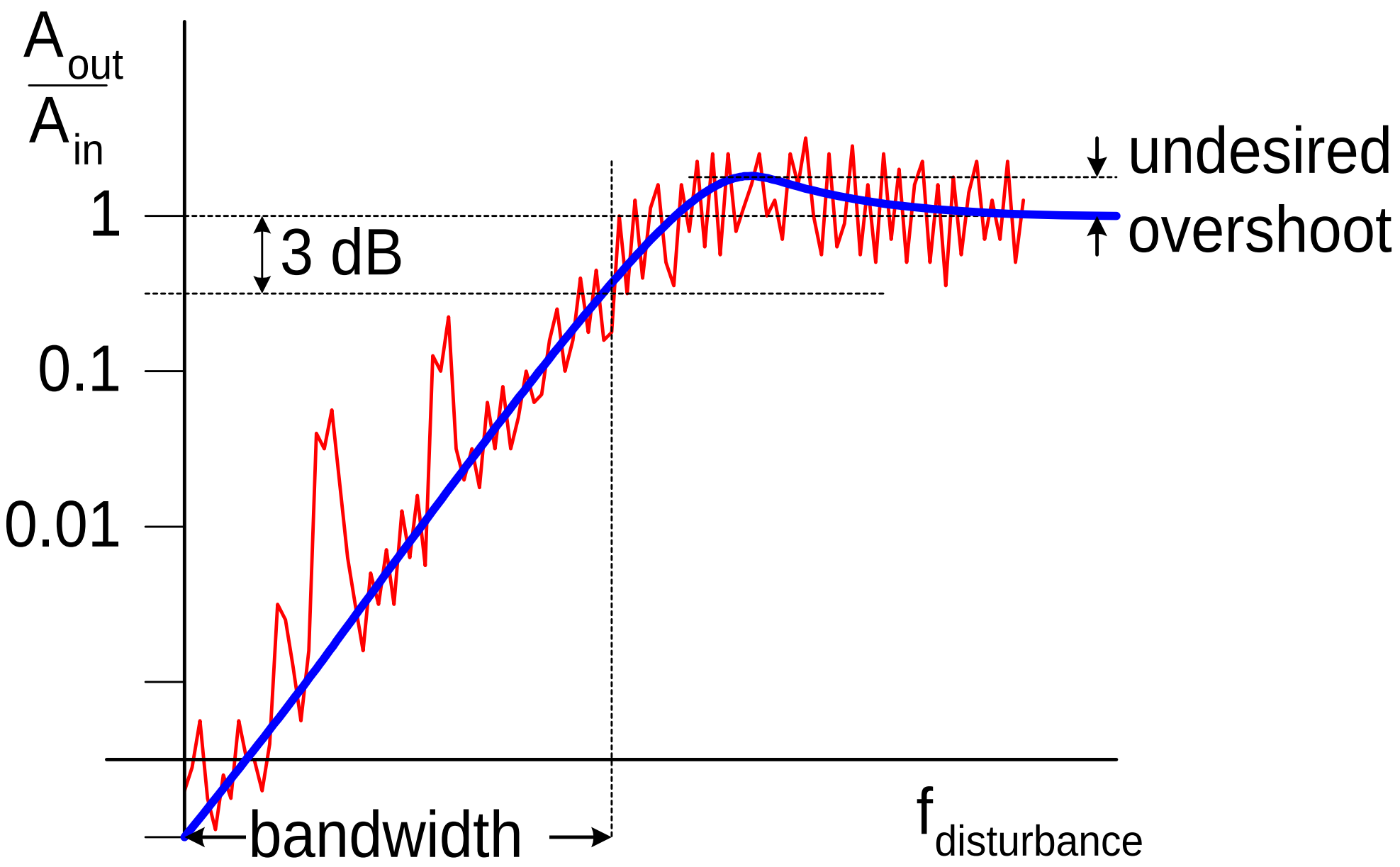
Block Diagram Control Measurement



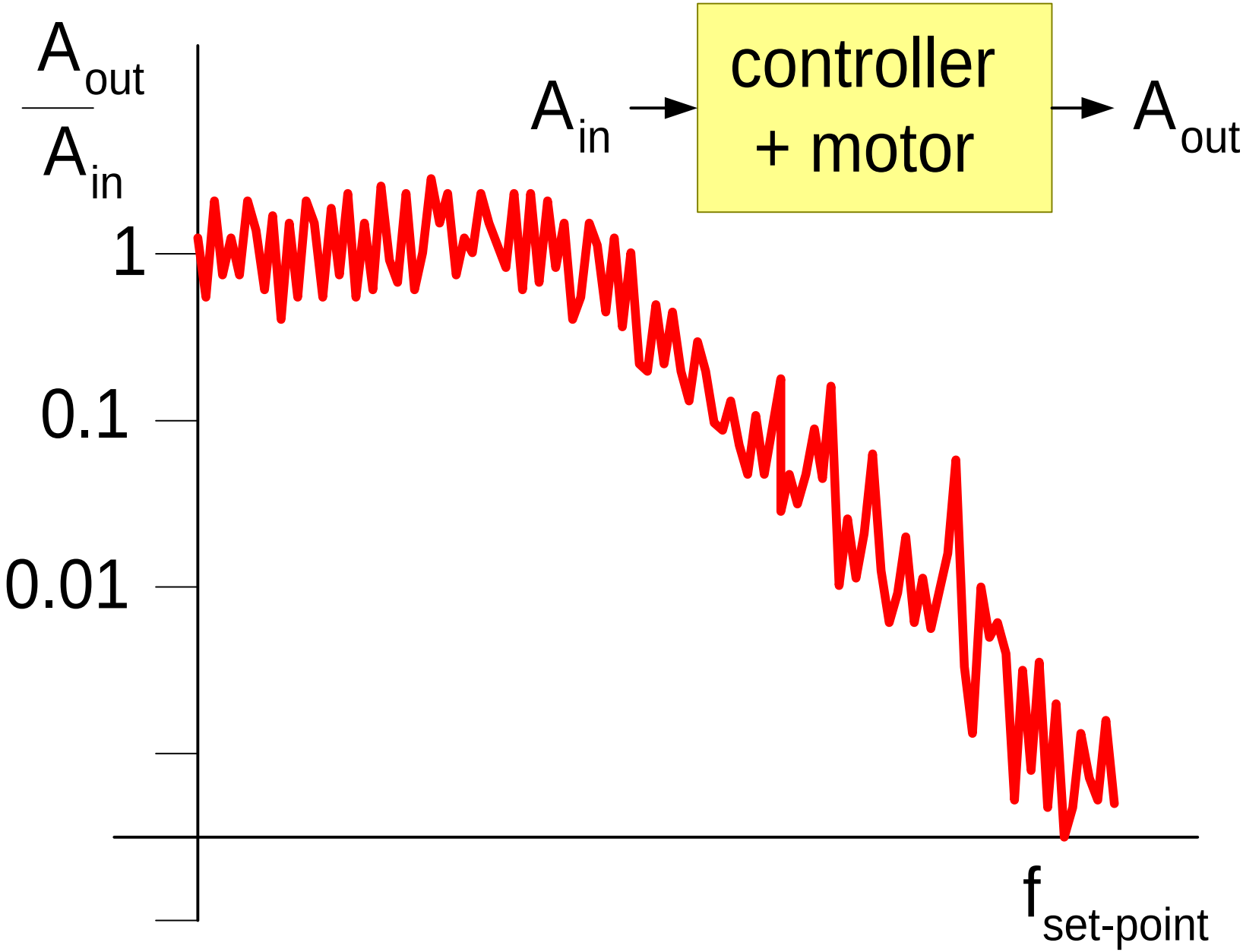
Measuring Disturbance Transfer



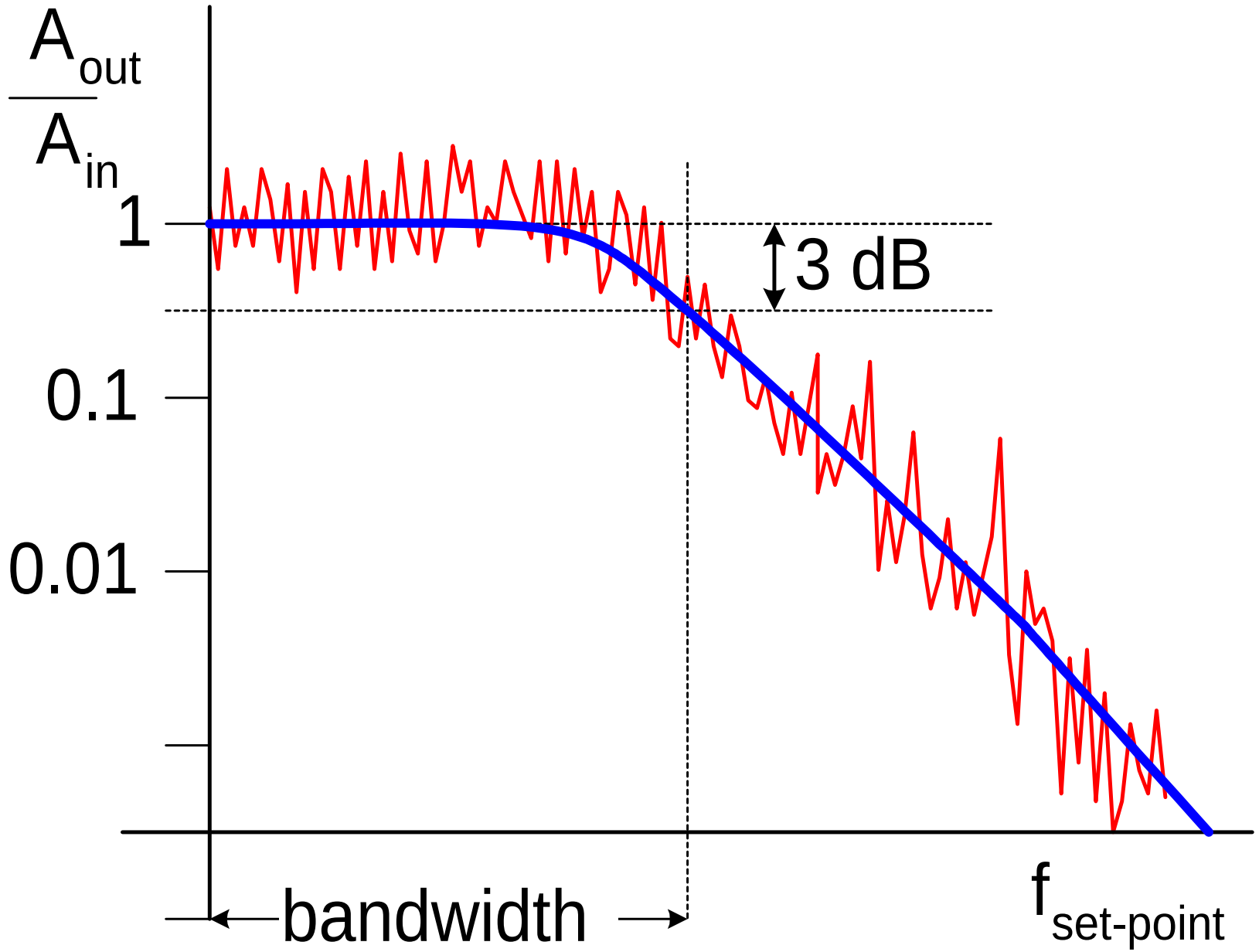
Idealized Disturbance Transfer



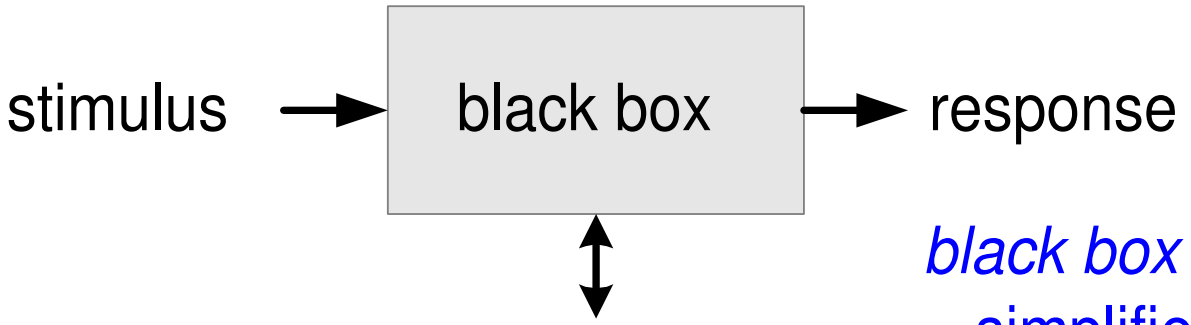
Measuring Tracking Response



Idealized Tracking Responce



Black Box Model



black box :

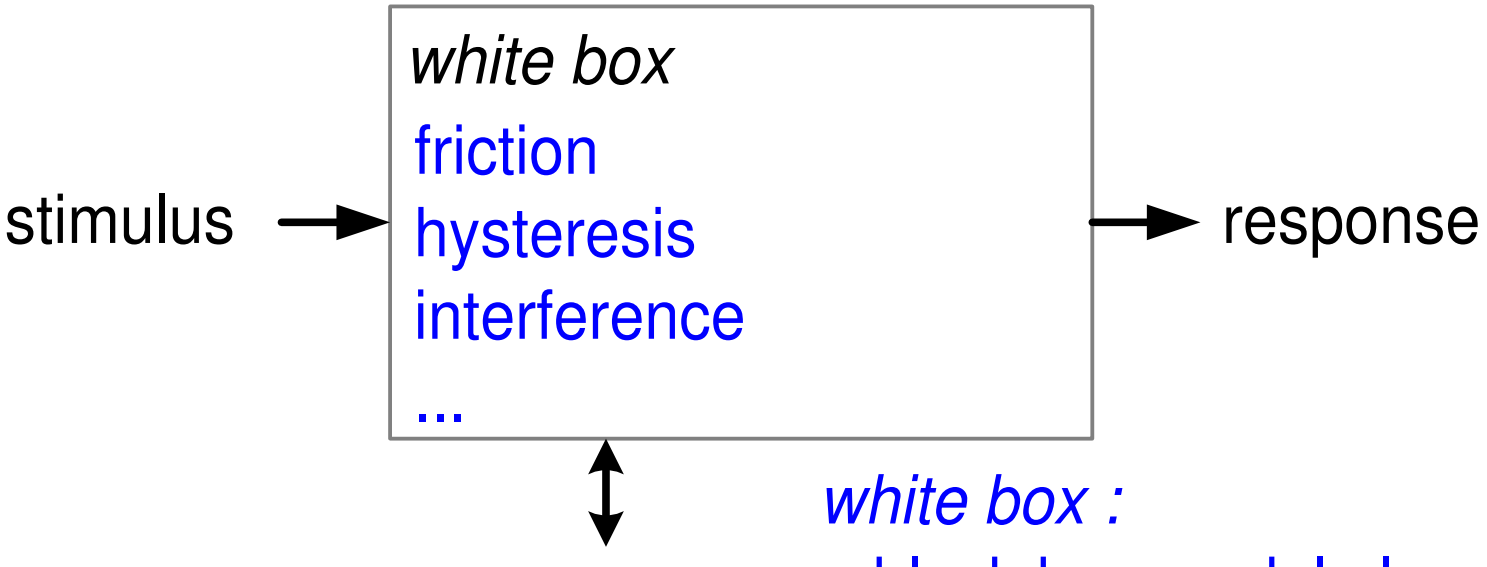
simplified model

mathematical formula

with physical interpretation:

response = f(stimulus, parameters)

White Box Model

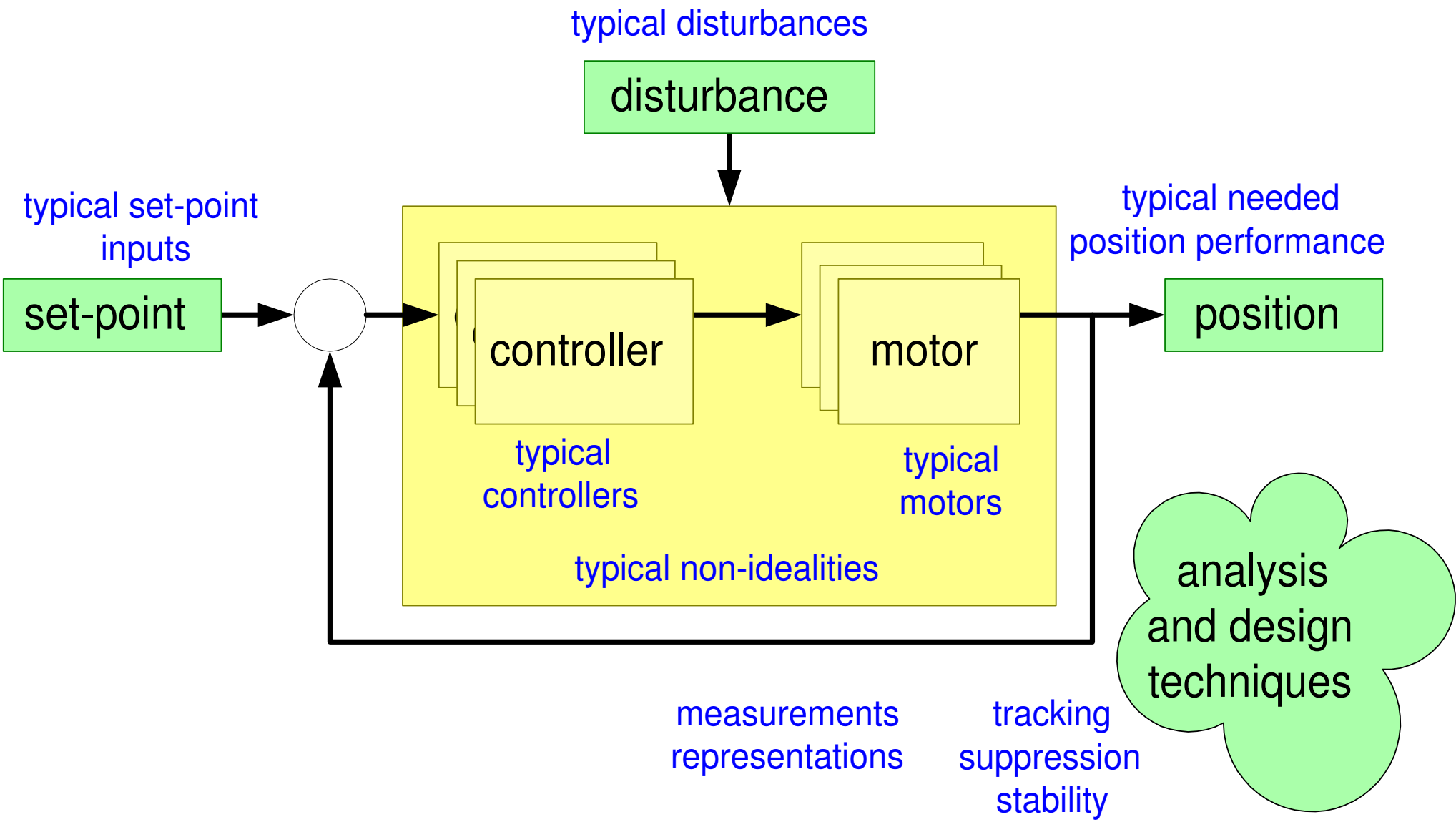


white box :

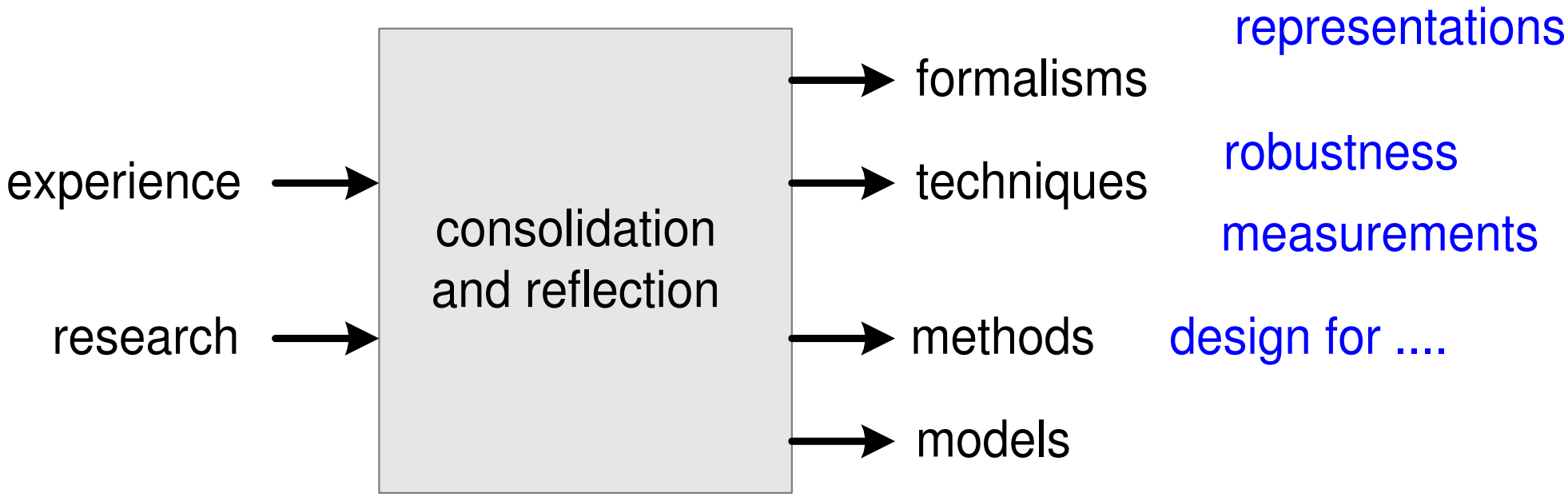
black box model plus
experience based reasoning
over non-idealities
response = f(stimulus, parameters,
some parametrized non-idealities)

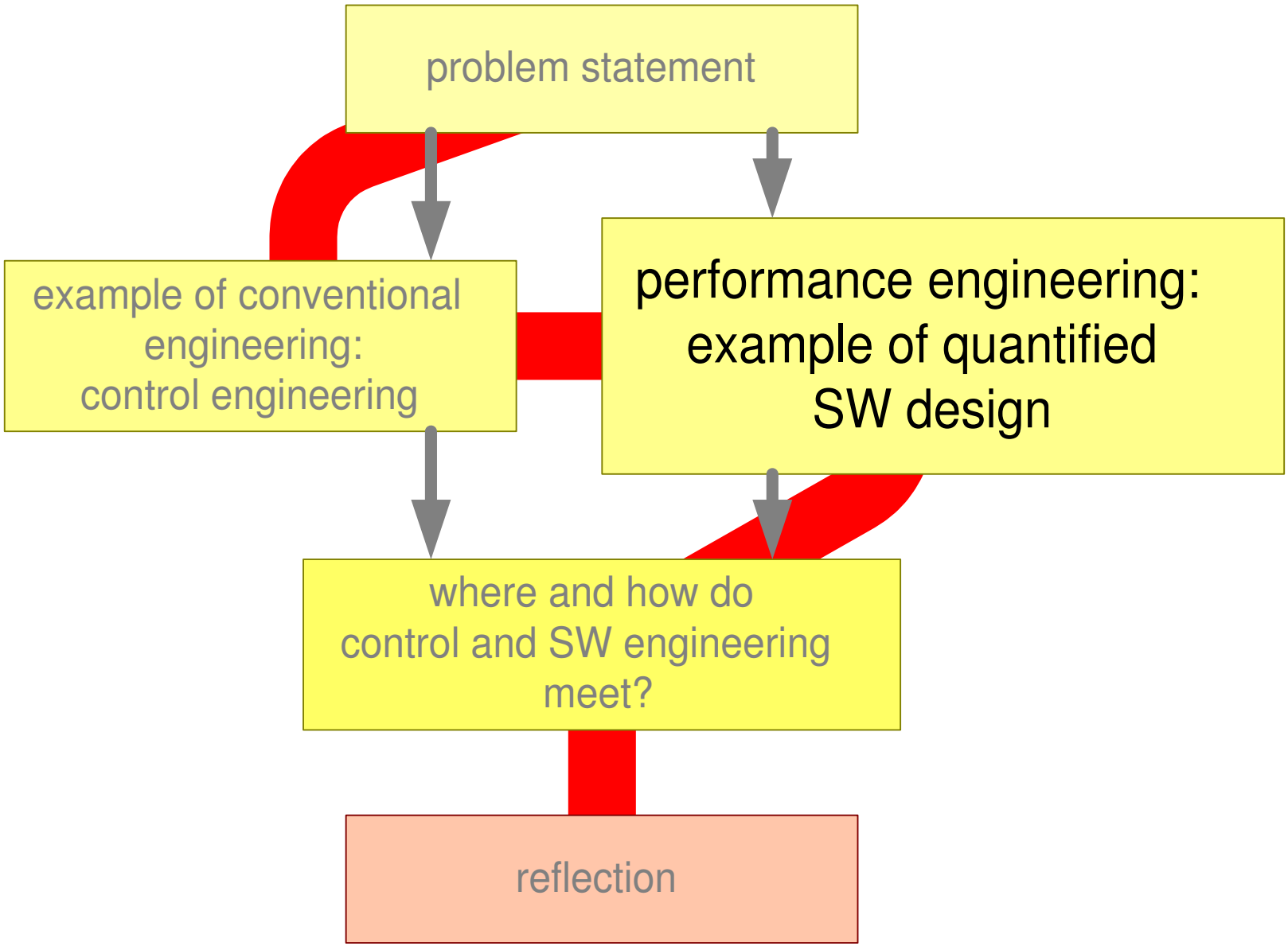
challenge: to know what non-idealities to ignore and to ignore as much as possible

Control Engineering Knowledge



Summary of Control Engineering




problem statement
example of conventional engineering: control engineering
performance engineering: example of quantified SW design
where and how do control and SW engineering meet?
reflection

What is the Performance of this Code?

application need:

at event 3*3 show 3*3 images instanteneous

design

design

Sample application code:

```
for x = 1 to 3 {
    for y = 1 to 3 {
        retrieve_image(x,y)
    }
}
```

or

alternative application code:

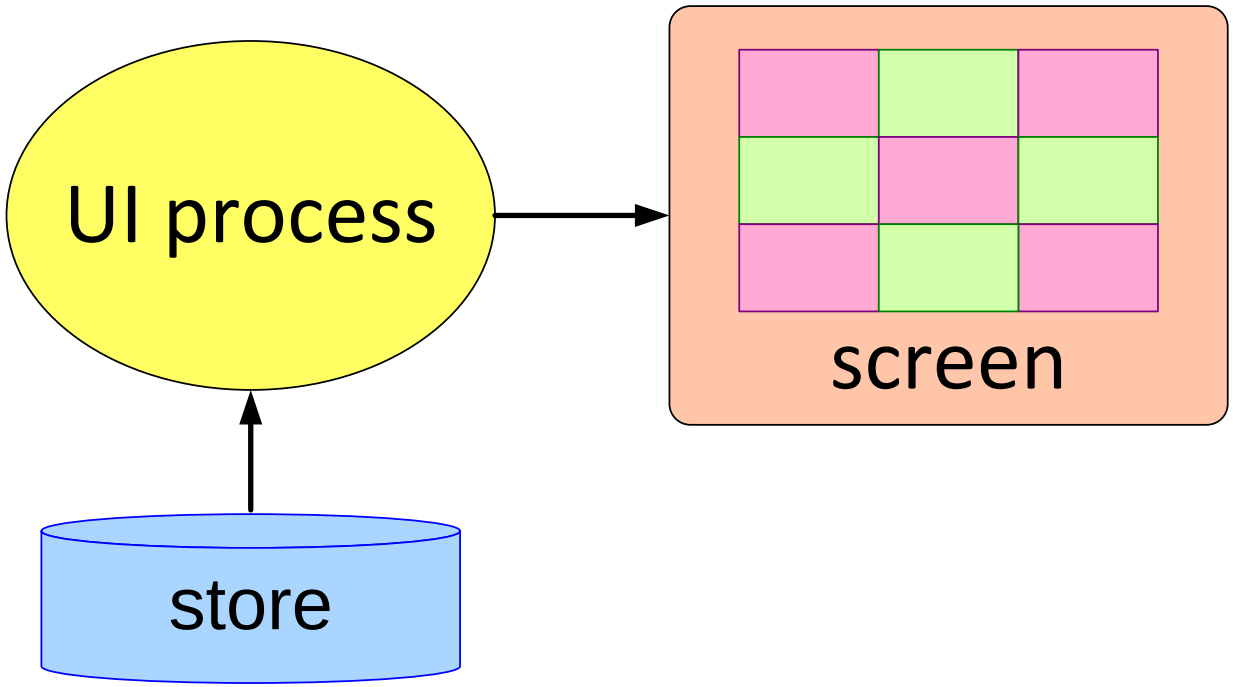
event 3*3 -> show screen 3*3

```
<screen 3*3>
    <row 1>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 1>
  <row 2>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 1>
  <row 2>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 3>
</screen 3*3>
```

What If....

Sample application code:

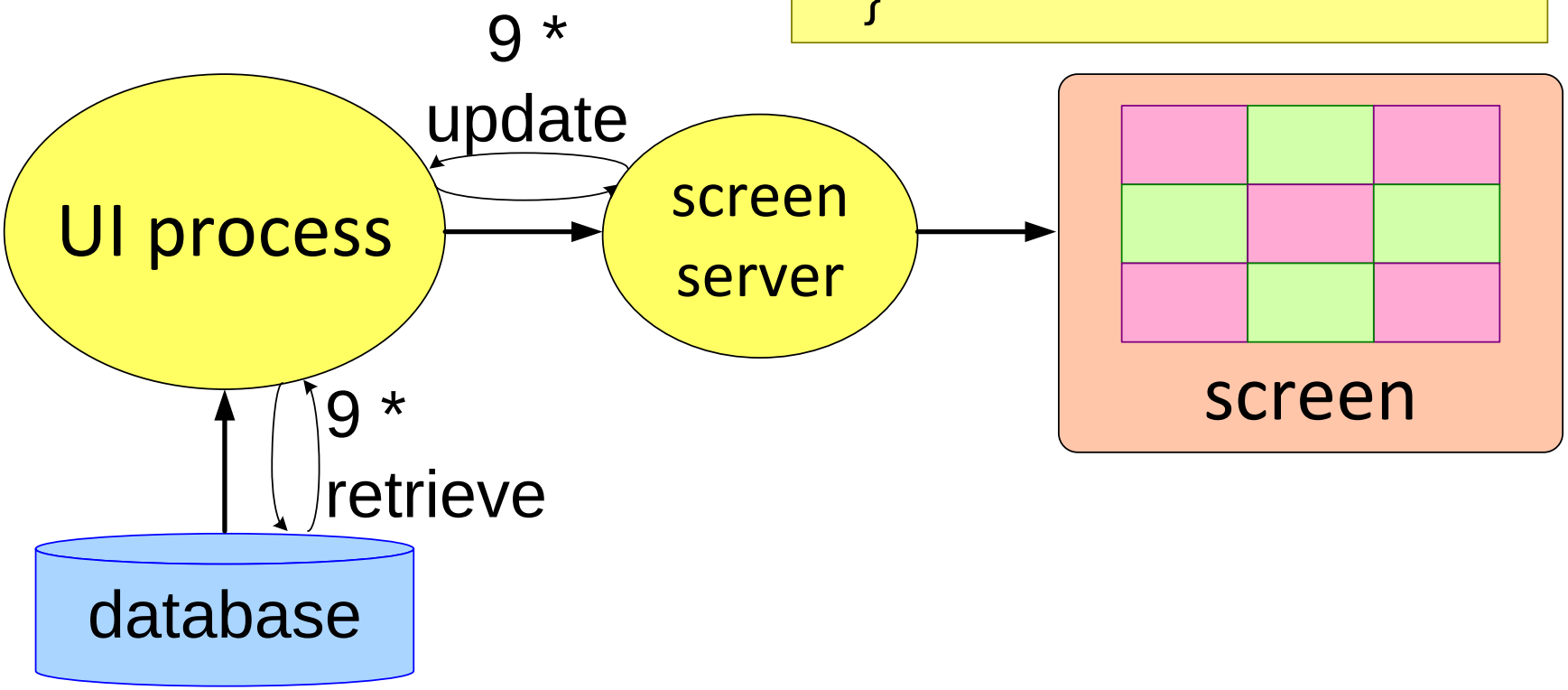
```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```



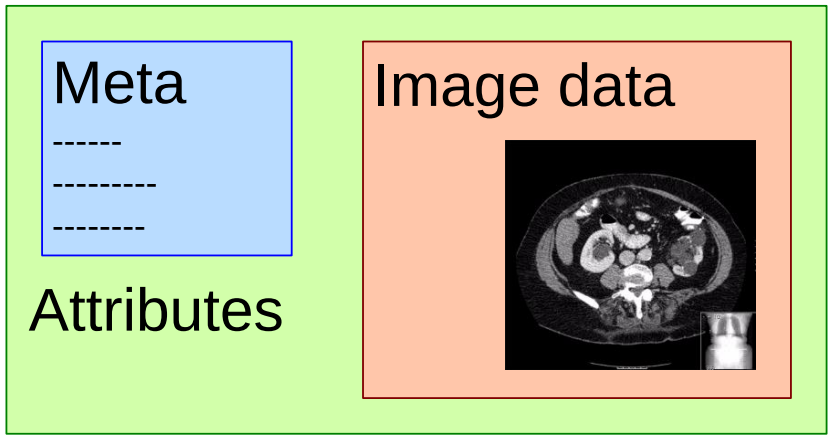
What If....

Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```



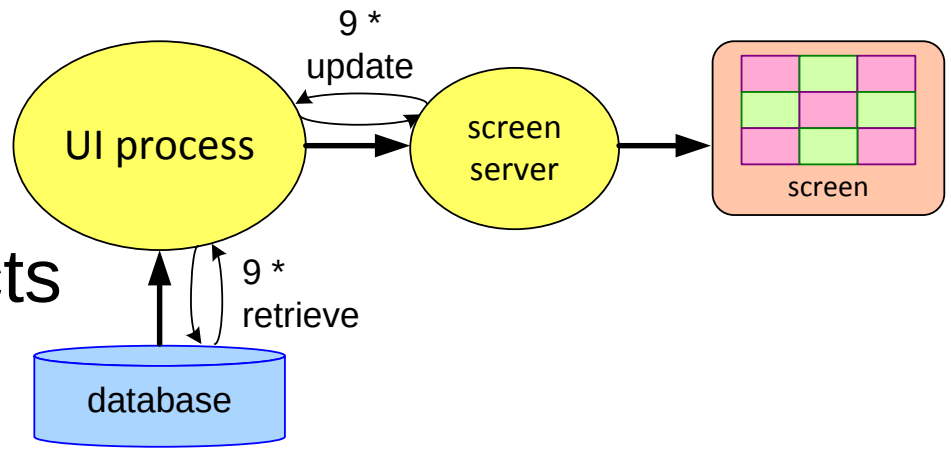
What If....



Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

Attribute = 1 COM object
100 attributes / image
9 images = 900 COM objects
1 COM object = 80µs
9 images = 72 ms



What If....

Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

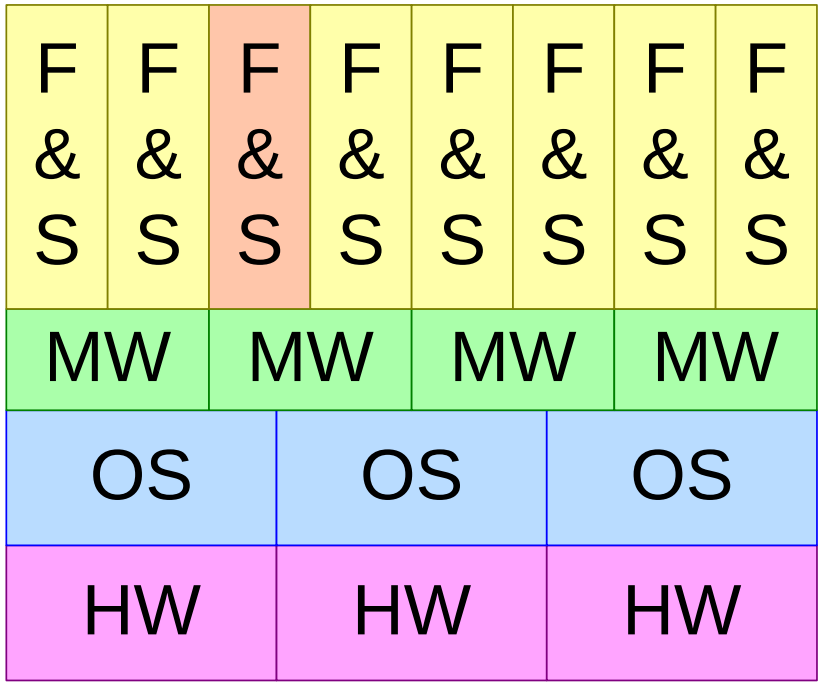
- I/O on line basis (512^2 image)

$$9 * 512 * t_{I/O}$$

$$t_{I/O} \sim= 1ms$$

- . . .

Challenge SW Performance Design



Functions & Services

Middleware

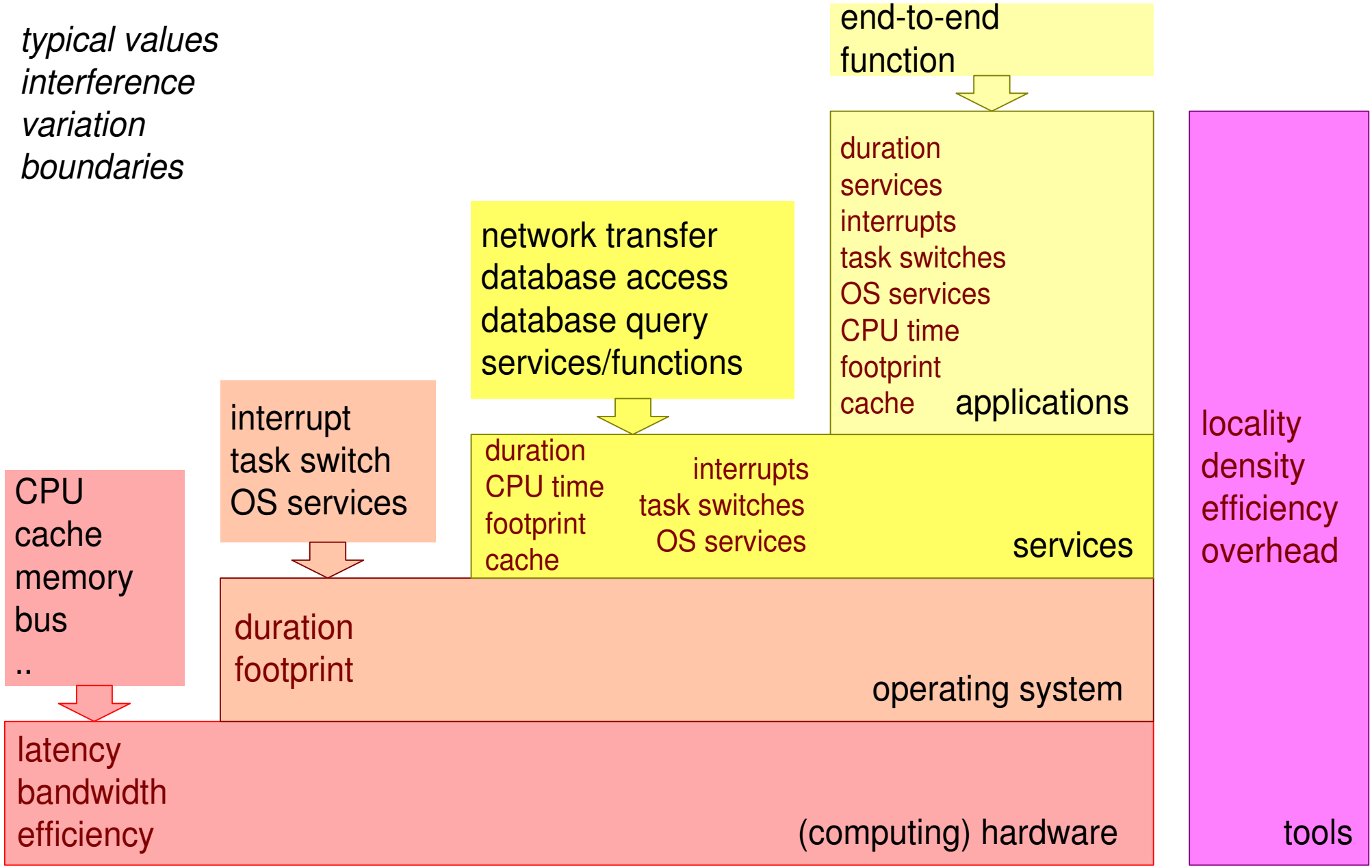
Operating systems

Hardware

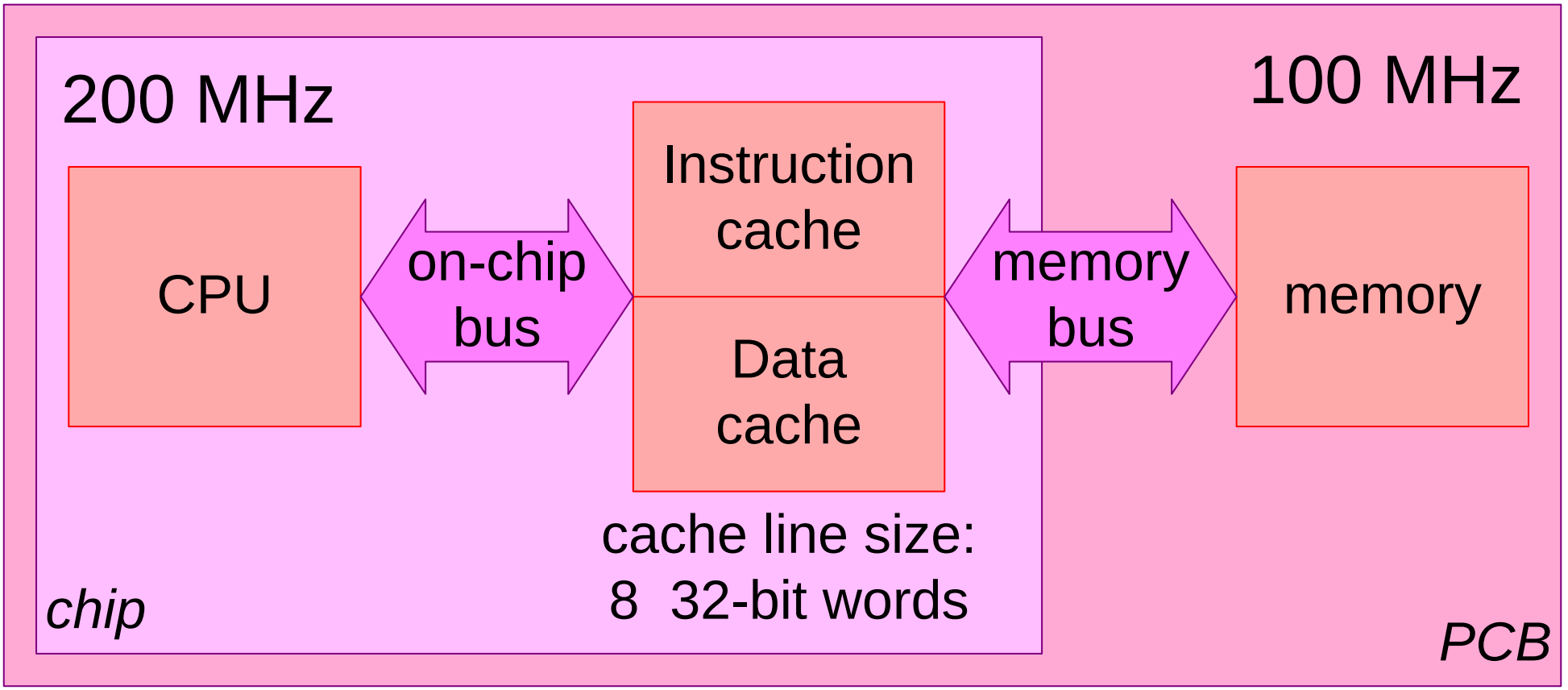
Performance = Function (F&S, other F&S, MW, OS, HW)
MW, OS, HW >> 100 Manyear : very complex

Challenge: How to understand MW, OS, HW
with only a few parameters

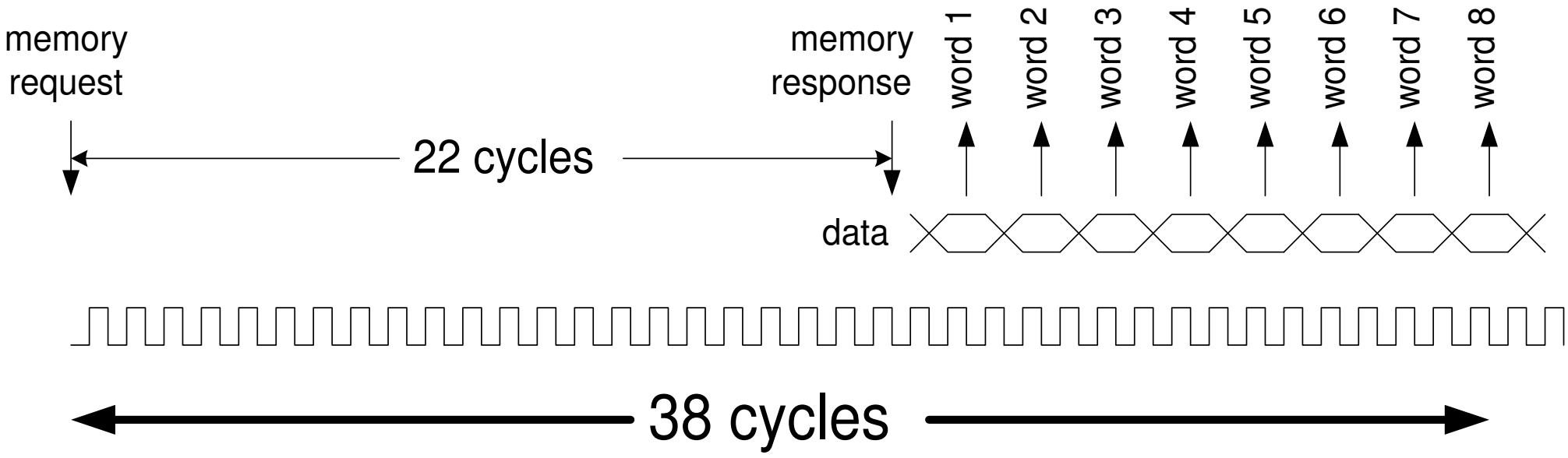
Layered Benchmarking



Case: ARM9 Cache Performance



Example Hardware Performance



memory access time in case of a cache miss
200 Mhz, 5 ns cycle: 190 ns

ARM9 200 MHz $t_{context\ switch}$ as function of cache use

cache setting	$t_{context\ switch}$
From cache	2 µs
After cache flush	10 µs
Cache disabled	50 µs

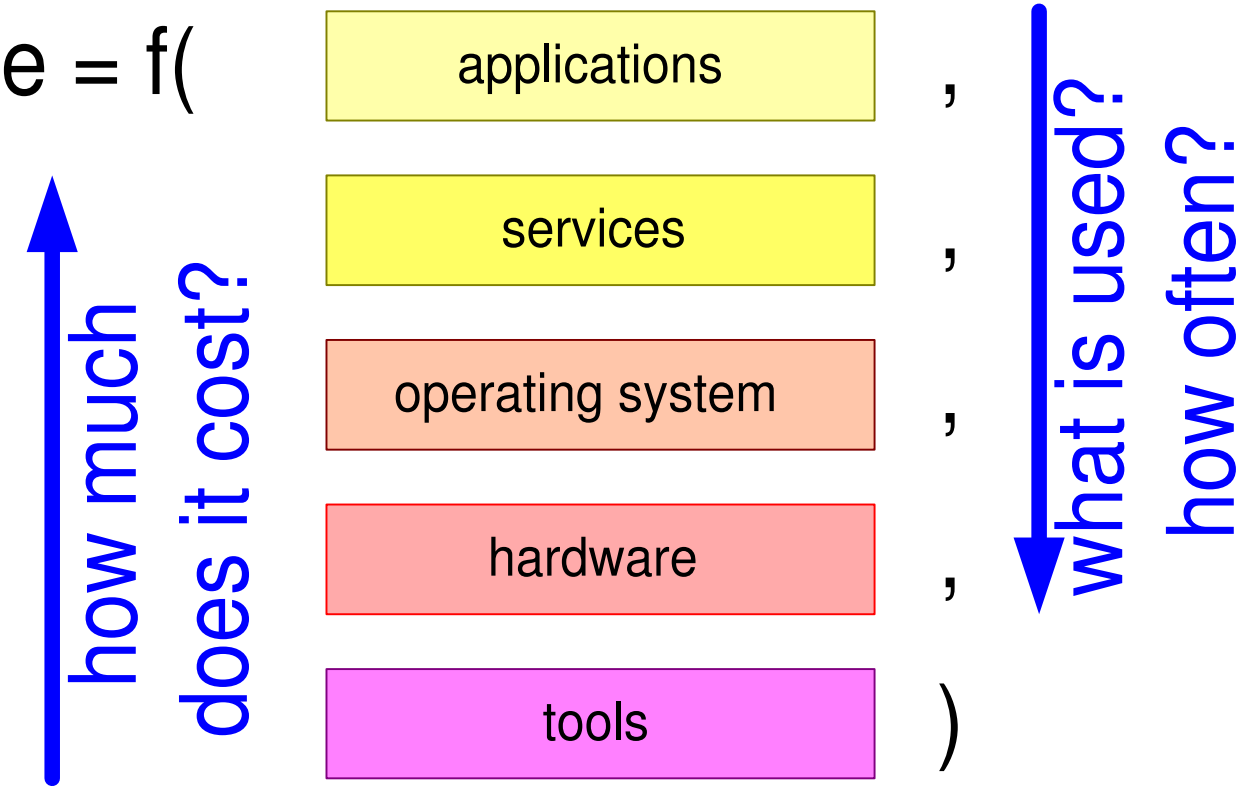
Context Switch Overhead

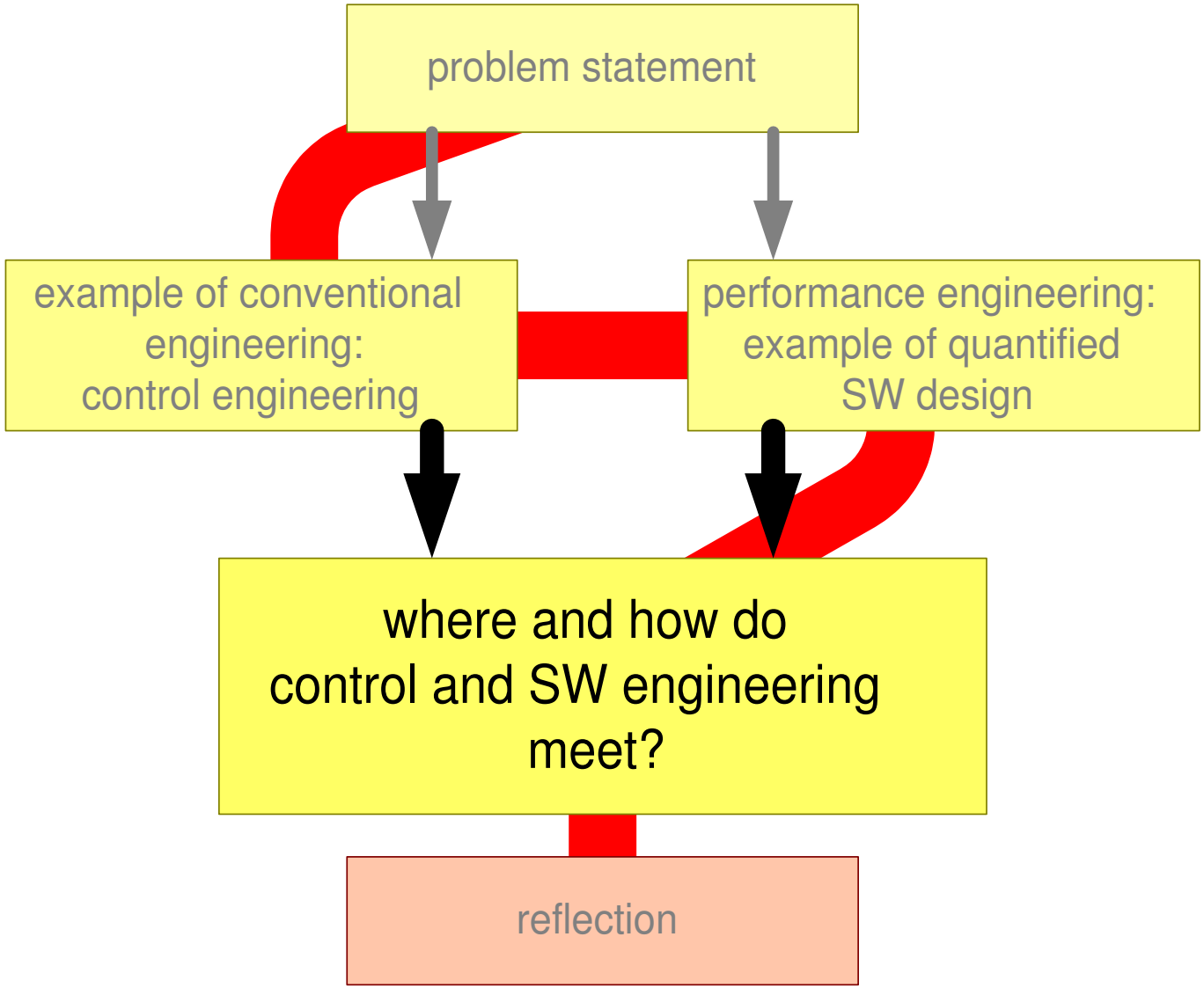
$$t_{overhead} = n_{context\ switch} * t_{context\ switch}$$

$n_{context\ switch}$ (s^{-1})	$t_{context\ switch}$ = 10µs		$t_{context\ switch}$ = 2µs	
	$t_{overhead}$	CPU load overhead	$t_{overhead}$	CPU load overhead
500	5ms	0.5%	1ms	0.1%
5000	50ms	5%	10ms	1%
50000	500ms	50%	100ms	10%

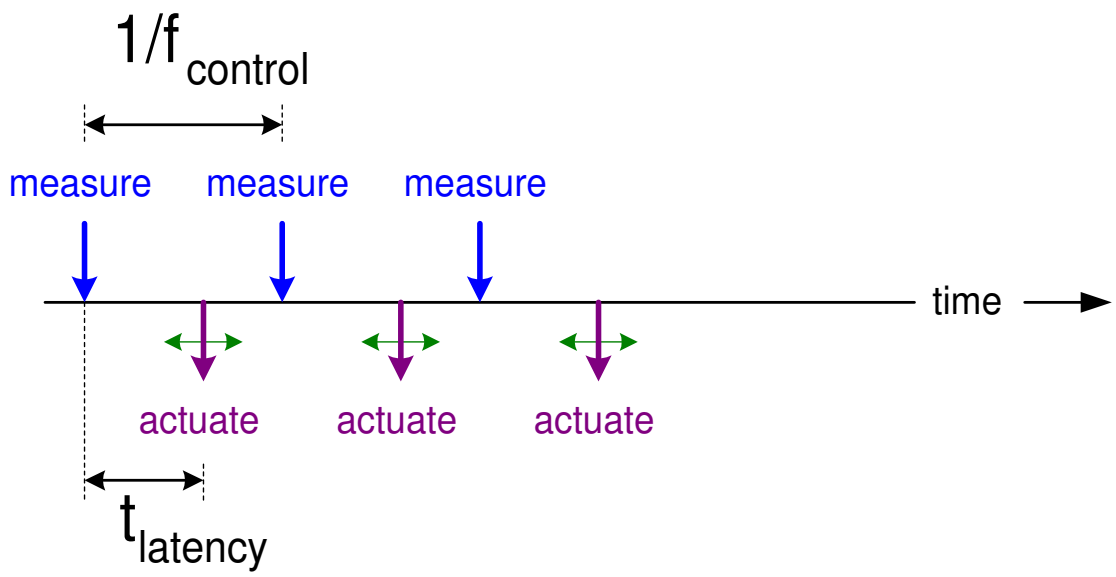
Performance as Function of all Layers

system performance = f(




problem statement
example of conventional engineering: control engineering
performance engineering: example of quantified SW design
where and how do control and SW engineering meet?
reflection

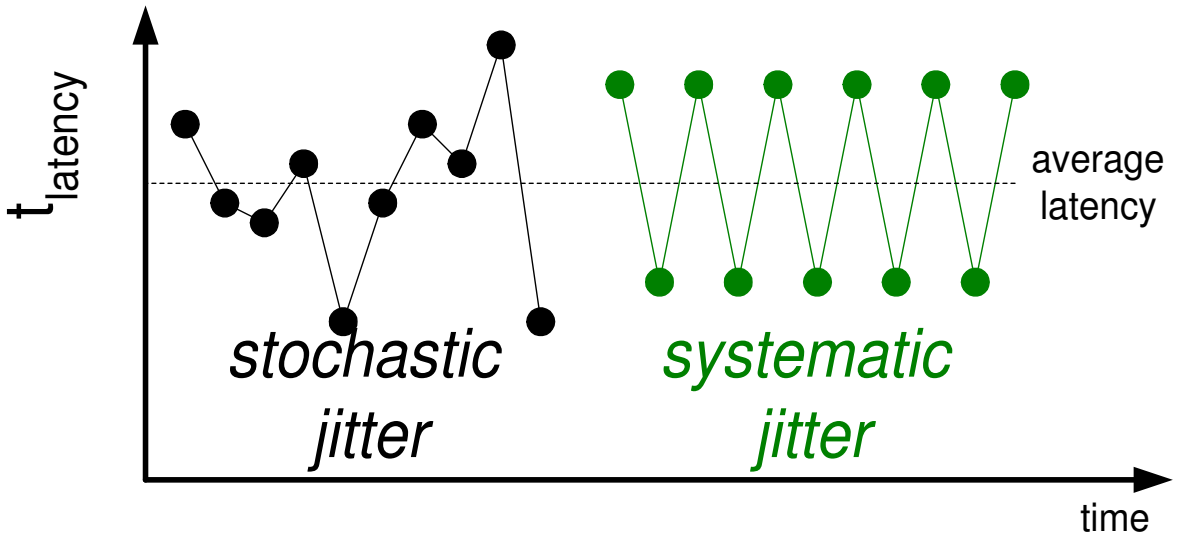
Impact of Timing on Control Performance



0e order

Performance $_{control}$ =

f ($f_{control}$, $t_{latency}$)

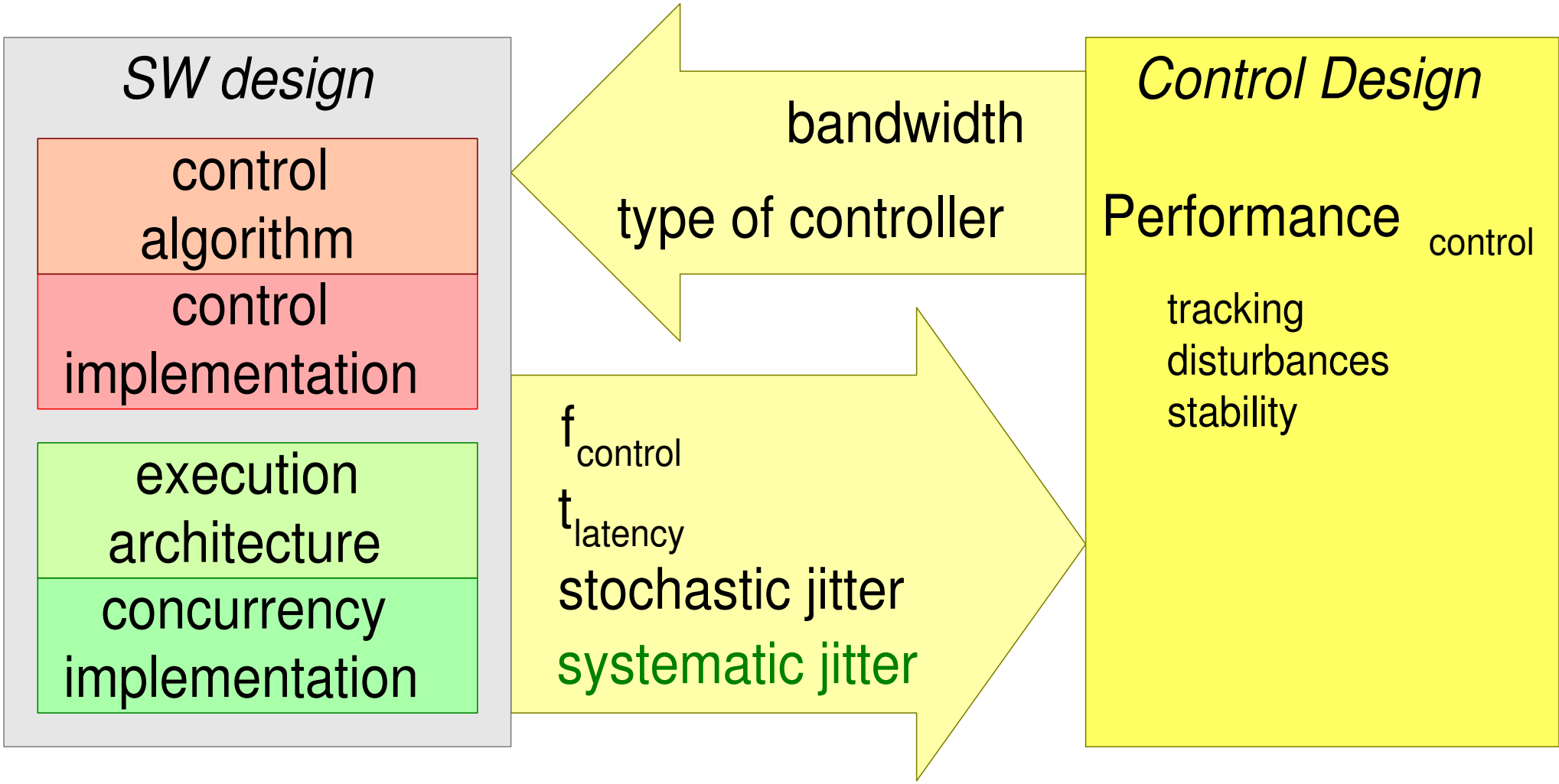


1e order

impact of jitter on

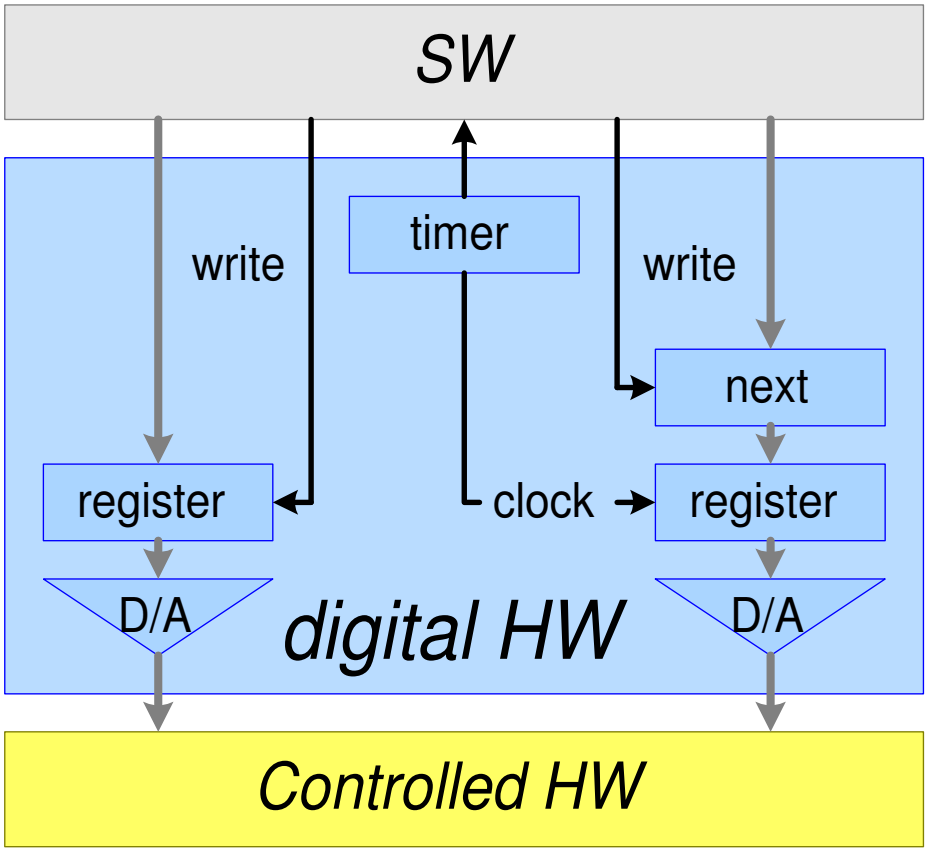
Performance $_{control}$

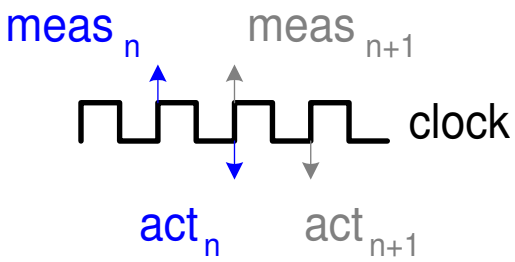
Mutual Impact of SW and Control Design



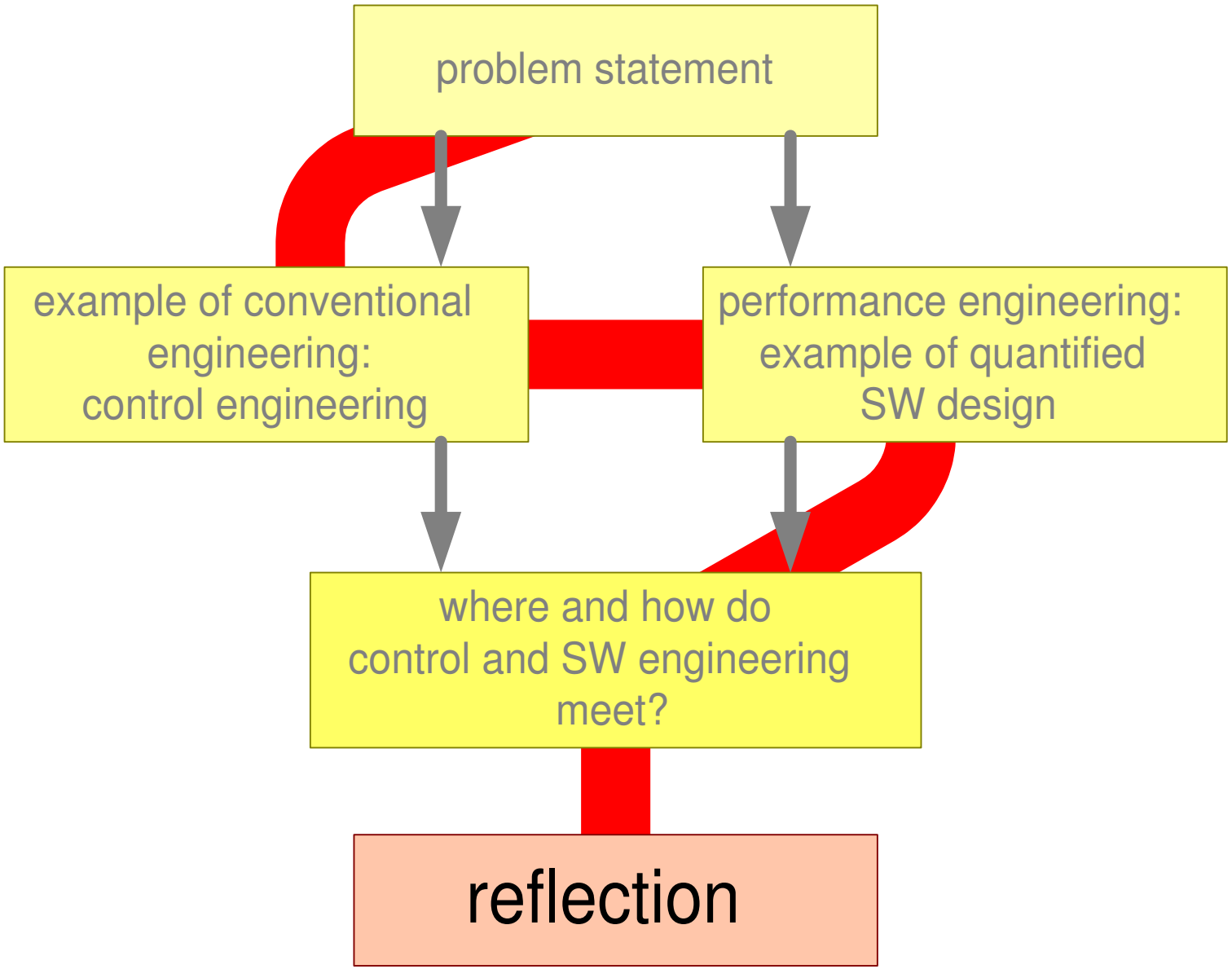
Impact of digital HW on SW and Control

single buffered
SW impl. jitter
variable latency

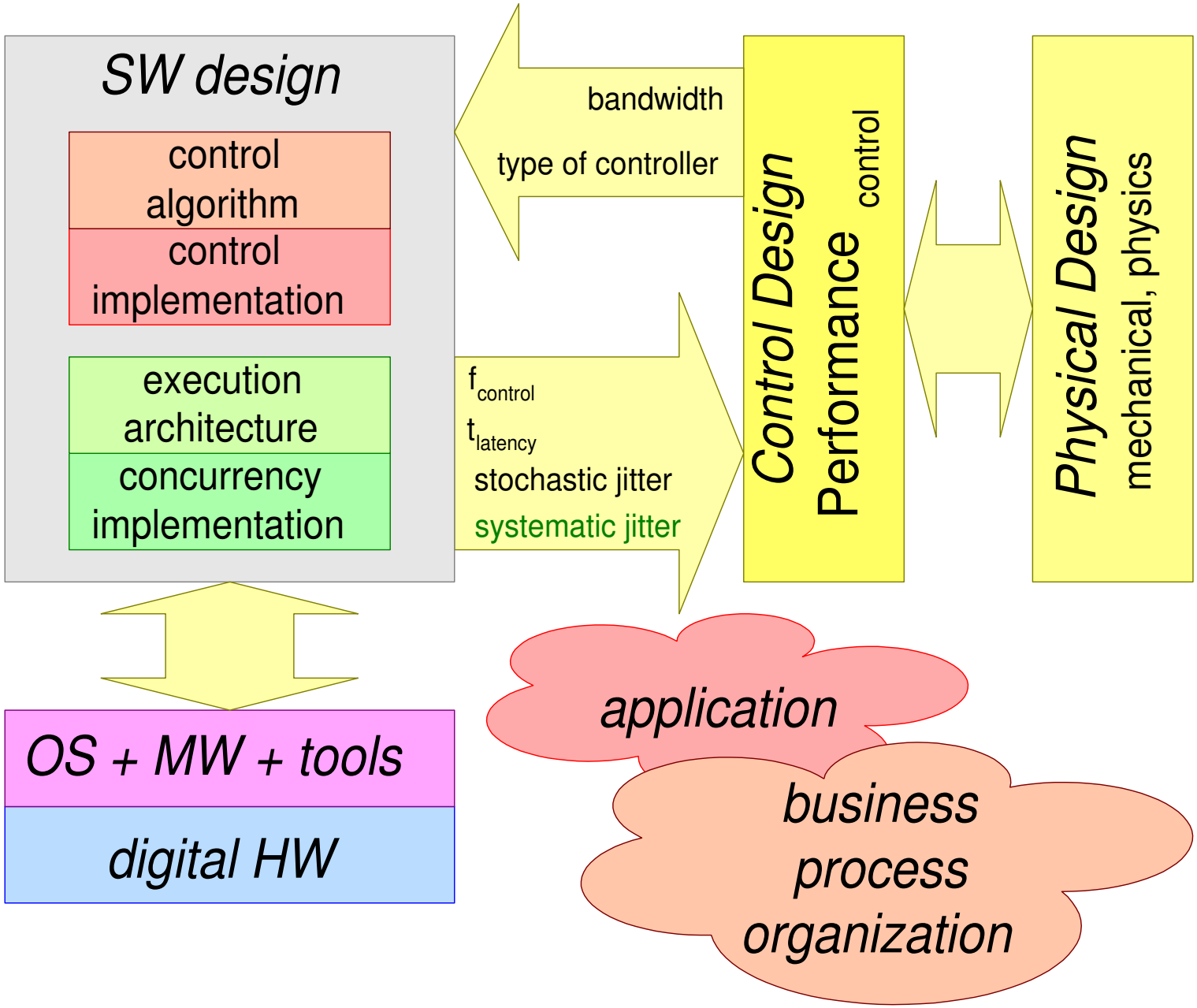




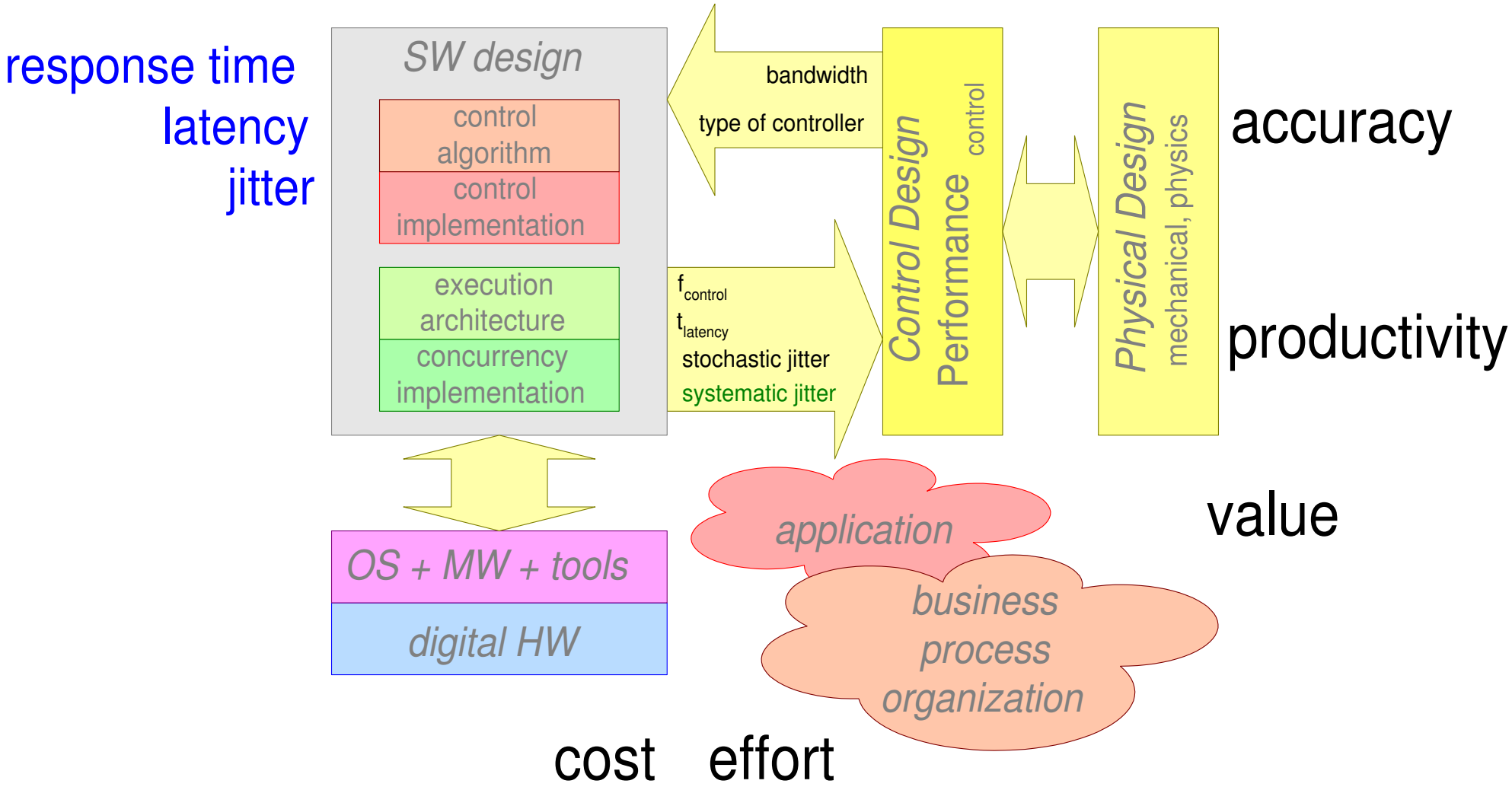
double buffered
jitter free
fixed longer latency


problem statement
example of conventional engineering: control engineering
performance engineering: example of quantified SW design
where and how do control and SW engineering meet?
reflection

SW design = **Multi-** Multi- Disciplinary



Quantifications Connect Disciplines



Questions?

After this slide some more quantification examples and issues are shown

Examples of Quantification; Electronic Patient Record

Customer	**A**pplication	**F**unctional	**C**onceptual	**R**ealization
objectives figures of merit	typical quantities	critical specs	working ranges	critical numbers
patients/physician physician income success rate failure rate integral cost	# patients # physicians # exams/day # exams/patient # information/ patient	productivity response time capacity	# transactions # queries peak&average	network speed CPU speed memory size power consumption query duration transaction overhead

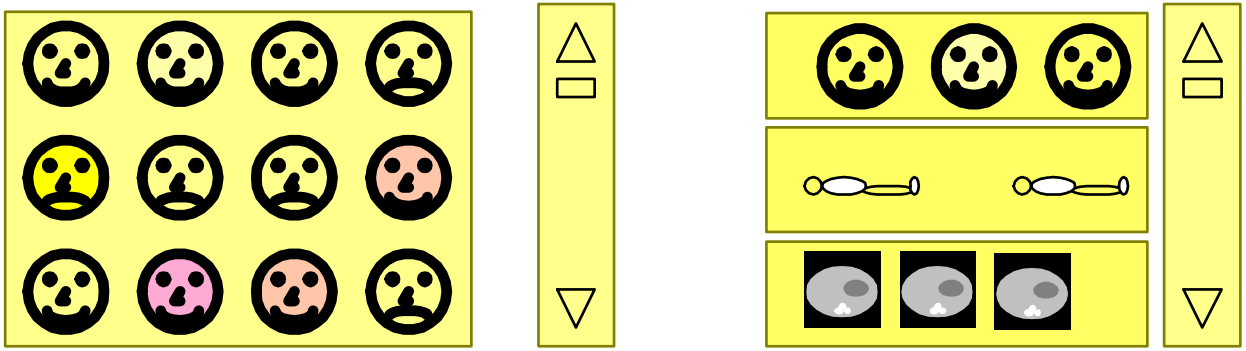
internal **O**perational view

market size	product life cycle	maintenance effort	# suppliers	effort	project size
market share	business model	update frequency	partners	cost	# engineers/discipline
growth rate	market segments	service crew	competitors	time	# teams

Where and When to Quantify

requirements analysis	paradigm boundaries application relevance design sensitivity
ranges and relations	typical, best, worst case dependencies
variation analysis	random vs systematic types of systematic variation time-base, rate of change
propagation analysis	amplification or dimming
evolution	application, business evolution technology evolution scaling, scaling boundaries

Example UI paradigms for Pictorial Index

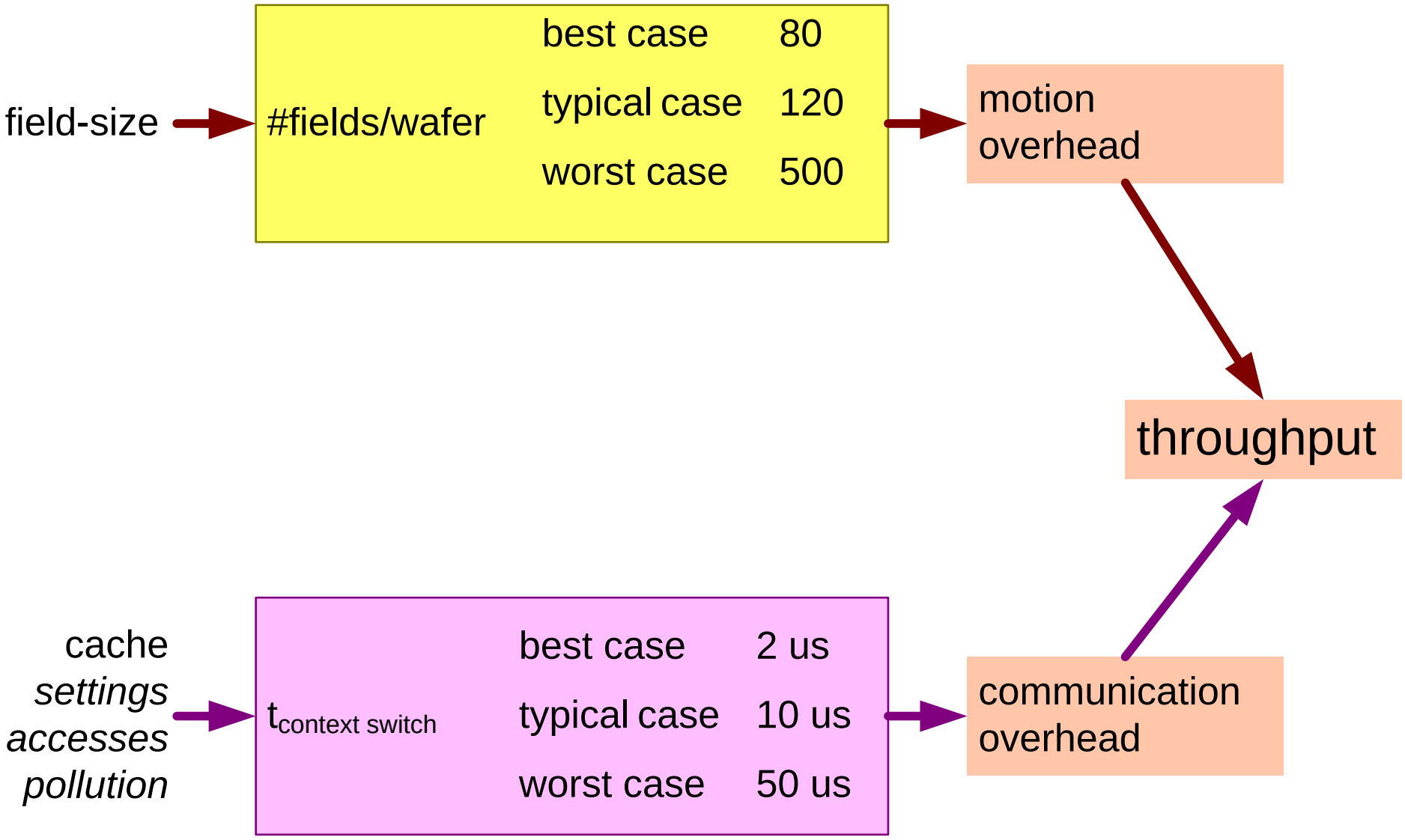


meta-information directory

show all	add scrolling	add hierarchy	meta-information search

1 10 10^2 10^3 10^4 10^5 10^6

Example of Combining Heterogeneous Quantifications



Example Computer Crime quantification

http://www.usatoday.com/tech/news/computersecurity/infotheft/2006-10-11-cybercrime-hacker-forums_x.htm

* $67.2 billion: FBI estimate of what U.S. businesses lose annually because of computer-related crimes.
* $8 billion: Consumer Reports estimate of what U.S. consumers lost the past two years because of viruses, spyware and Internet scams.
* 93.8 million: Privacy Rights Clearinghouse's count of personal records reported lost or stolen since February 2005.
* 26,150: The Anti-Phishing Working Group's count of unique variations of phishing scams reported in August 2006.

Typical costs of goods and services in forums:

* $1,000 to $5,000: Trojan program that can transfer funds between online accounts.
* $500: Credit card number with PIN.
* $80 to $300: Change of billing data, including account number, billing address, Social Security number, home address and birth date.
* $150: Driver's license.
* $150: Birth certificate.
* $100: Social Security card.
* $7 to $25: Credit card number with security code and expiration date.
* $7: PayPal account log-on and password.
* 4% to 8% of the deal price: Fee to have an escrow agent close a complex transaction.
* Free: Access to a service that gives details of the issuing bank for any credit card number.

1 -- Representative asking prices found recently on cybercrime forums

Source: USA TODAY research
referenced by http://groups.google.co.in/group/control-computer-crimes/