SARC = *Semi Analytical Recursive Convolution*

This simulation approach is dedicated to *Linear Time Invariant* analog circuit (`LTI`)

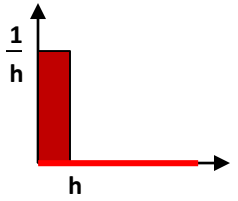
There is *no convergence* nor enigmatic user setup parameters (*since there is no parameter !*)

The *step calculation* is directly the result of *matrix arithmetic* ***sequential*** *operations*

Moreover, an *estimation* of the *step interpolation error* can be evaluated as a possible ***metric*** of the simulation *quality*

Inductances and *Capacitances* values can be changed "*on the fly*"

thanks to the algorithm conservation of the *flux* or the *charge* at the ***instant*** of the changes (called *percussion*)

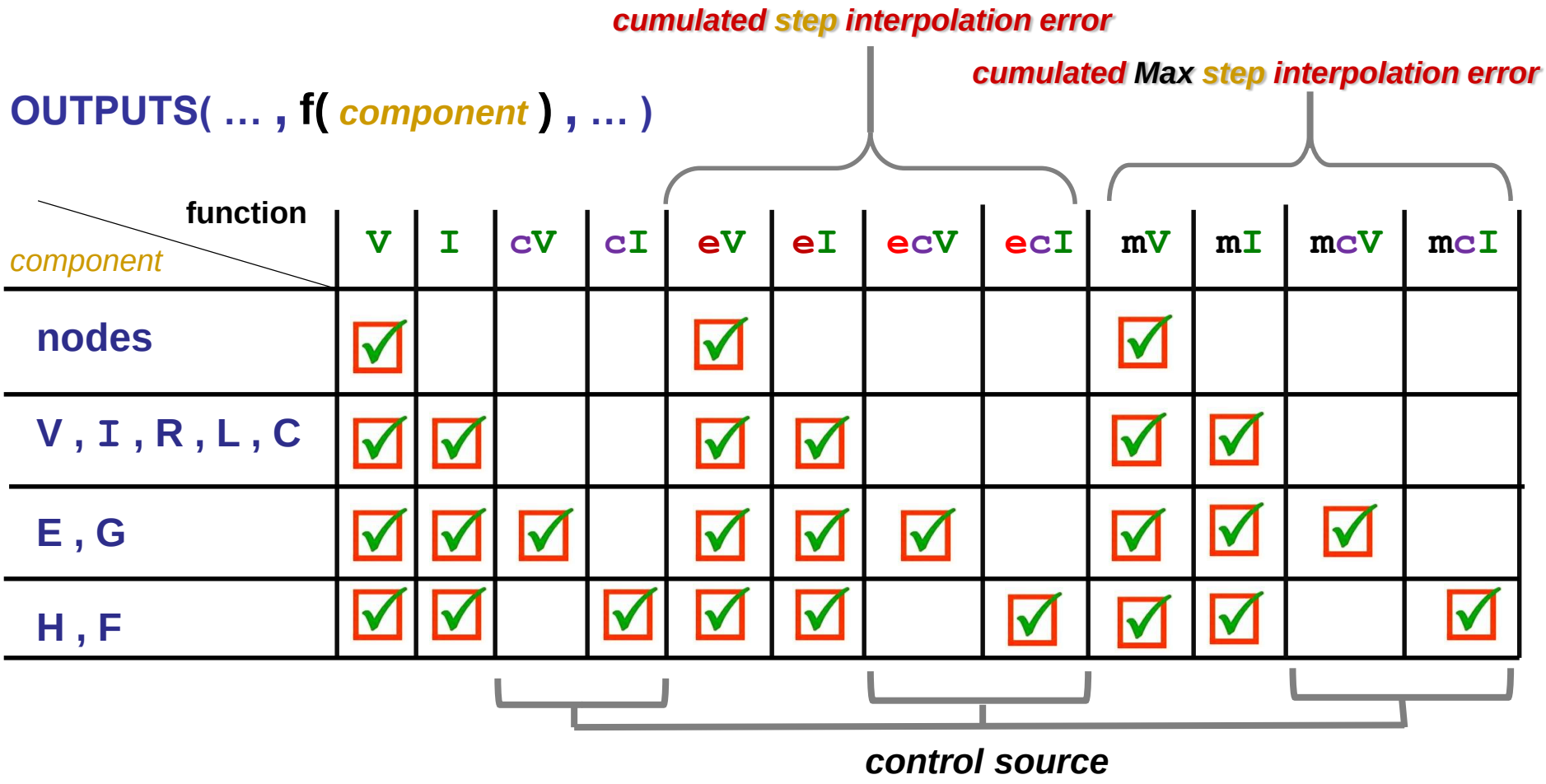
Accurate *impulsional response* can be simulated (using the input 

)

SARC is able to return to the user the *numerical values of the poles* as well as the overall value of the *energy* accumulated within the L and C

Finally, symbolic *Laplace transfer functions* of the circuit can also be returned

Maxima output syntax



Moreover, the following declaration is also possible in **OUTPUTS(…)**

J( ) **that stands for overall L & C *stored energy calculation* *(Joules)***

The following *user request* can be added in the ***Maxima*** circuit description

PRINTPOLES(N) ; **print *(up to* N *times*) the *POLES* of the system matrix "A"**

During **SARC** execution, whenever some **R , L** or **C**

are ***"percuted"*** the "**A**" matrix gets modified, and its ***POLES*** printed as

real* part** ***imaginary* part** **=** ***frequency

that is more convenient than

the theoretical pulsation ω (=> 2 π f)

AKA(**Ccommon** , C1 , C2 , C3) ;

AKA(**Rcommon** , “5E-3” , R1 , R2 , R3) ;

AKA(**Rcommon** , R1, R2 , R3 , “1500”) ;

SARC is based on a (*semi*) *analytical* approach

⇨ it does not require any "*rule of thumb options*" to control its ***execution*** & ***accuracy***

However, at **Maxima** level , *the following* "***modes***" *can be selected*

rubmode (0 *by default*) to partially / totally disable the update of the ***output vector*** with the information at **t+**

- **rubmode : 1 ;** "omit" the update BUT only in case of L and/or C percussion (*glitch* < h *not shown*)
- **rubmode : 2 ;** constantly "omit" the update (thus ***immediate*** STEP ***changes*** *NEVER shown*)

It should be noted that, in any cases, the rest of the simulation remains accurate (except that outputs at **t+** may be *hidden*)

degenmode (true *by default*) to enable / disable the ***degeneration*** solver

- **degenmode : false ;** *disable* the solver
  in case of degeneration, ***RESISTORS*** must be inserted in the NETLIST to resume execution

sdegenmode (true *by default*) to enable / disable the ***source degeneration*** solver

- **sdegenmode : false ;** *disable* the solver
  in case of degeneration, ***RESISTORS*** must be inserted in the NETLIST to resume execution

DEGENERATION can result from 2 different circuit configurations

- **a *shared voltage across* a group of capacitors**
  (or a *common current* for inductors)

  examples:

  // capacitors *serial* inductors capacitors *connected as polygons* **...**

- a *common current* constraining *flawlessly* a group of capacitors
  (or a *shared voltage across* inductors)

  This configuration is the same that reports floating nodes during SPICE analysis

  examples:

  serial capacitors // inductors

a third hassle exists that could also complicate the symbolic determination of matrix "A"

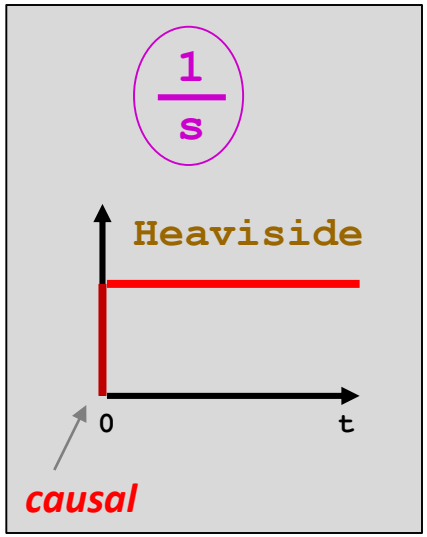
- **the first configuration but with an *input*** *(or* controlled source *)* ***as part of the group***

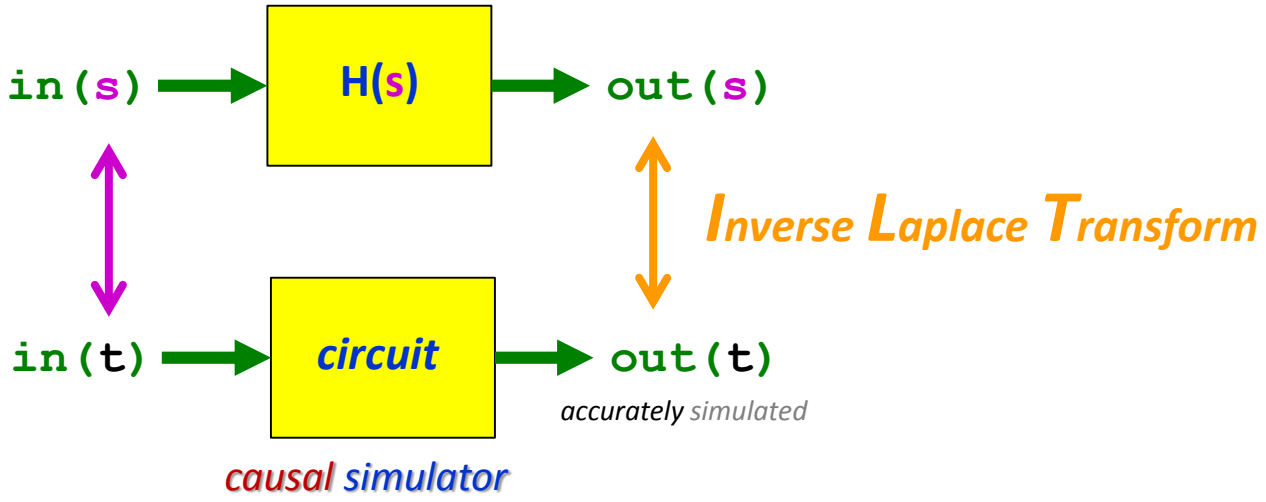
  *for **capacitors** with **driving voltage** or **inductors** with **driving current** this results in a **Dirac transfer** of energy to a **pure reactance bridge** (due to the absence of any **real world** **time constant**)*

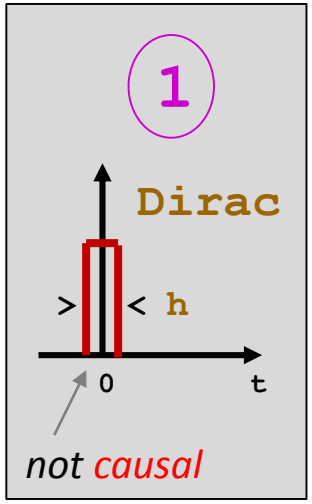
  Unlike the usual case , *it should be noted that the **state variables** associated to those bridges cannot be confounded with the **V** (or **I**) of **their corresponding components** (since a **direct** **source** **contribution** also exists)*

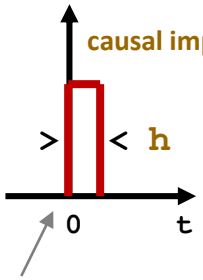
IDEALISATION <=> *is the **circuit** able to generate the **burst** ??? does the **reactance** able to survive to such **burst** ???*

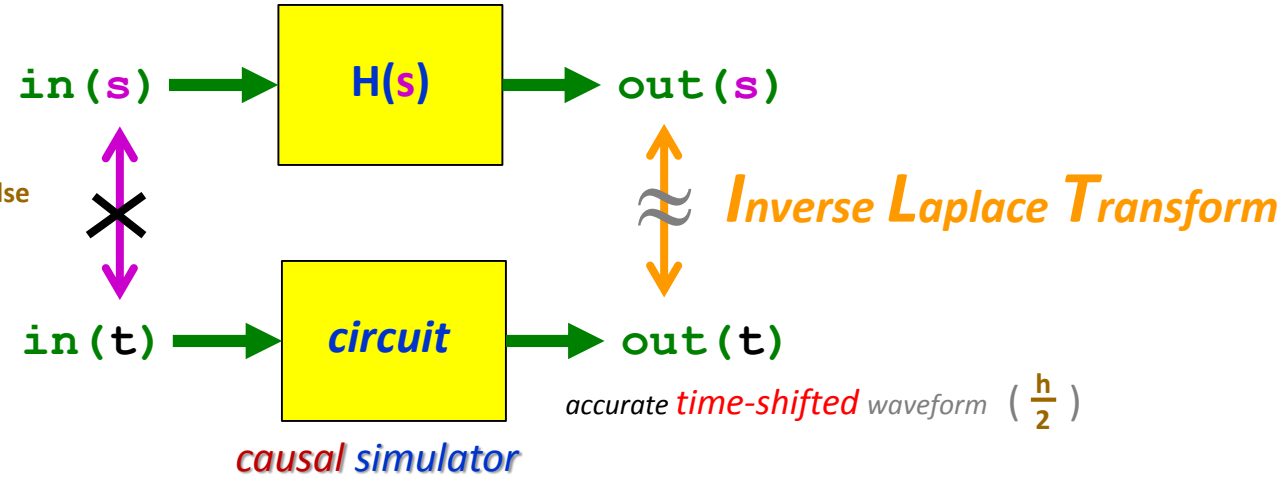
Validation : simulation versus Laplace

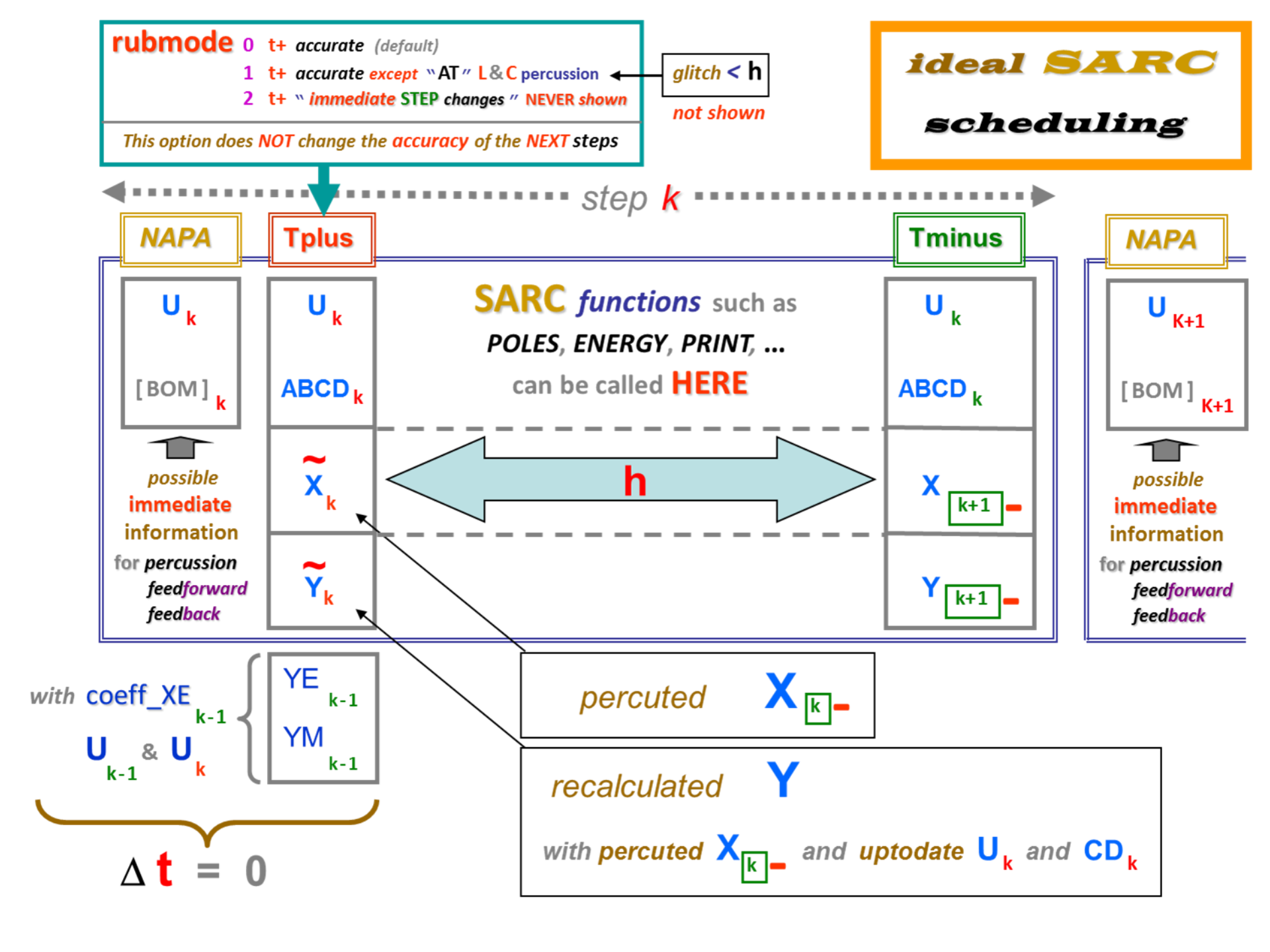


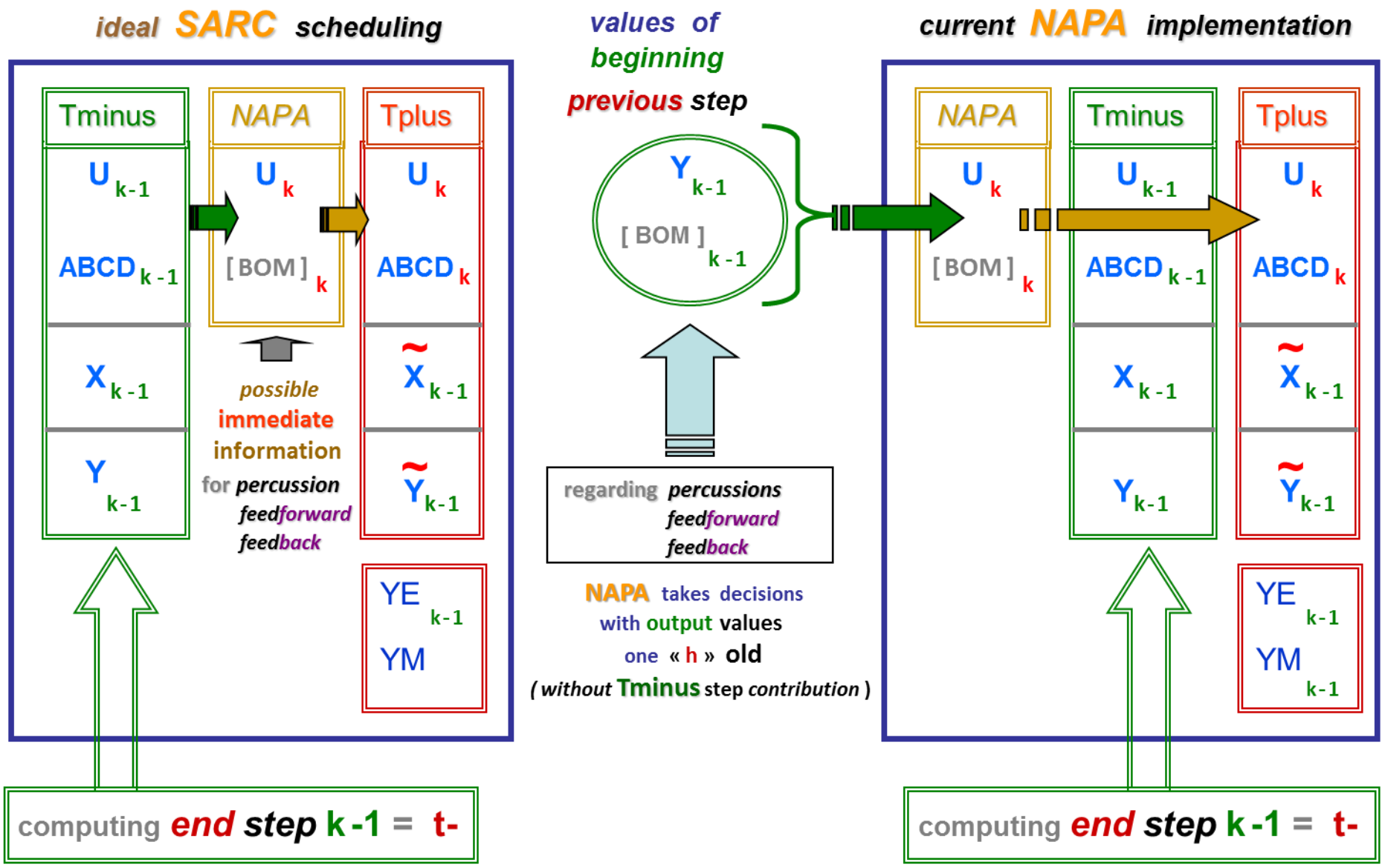




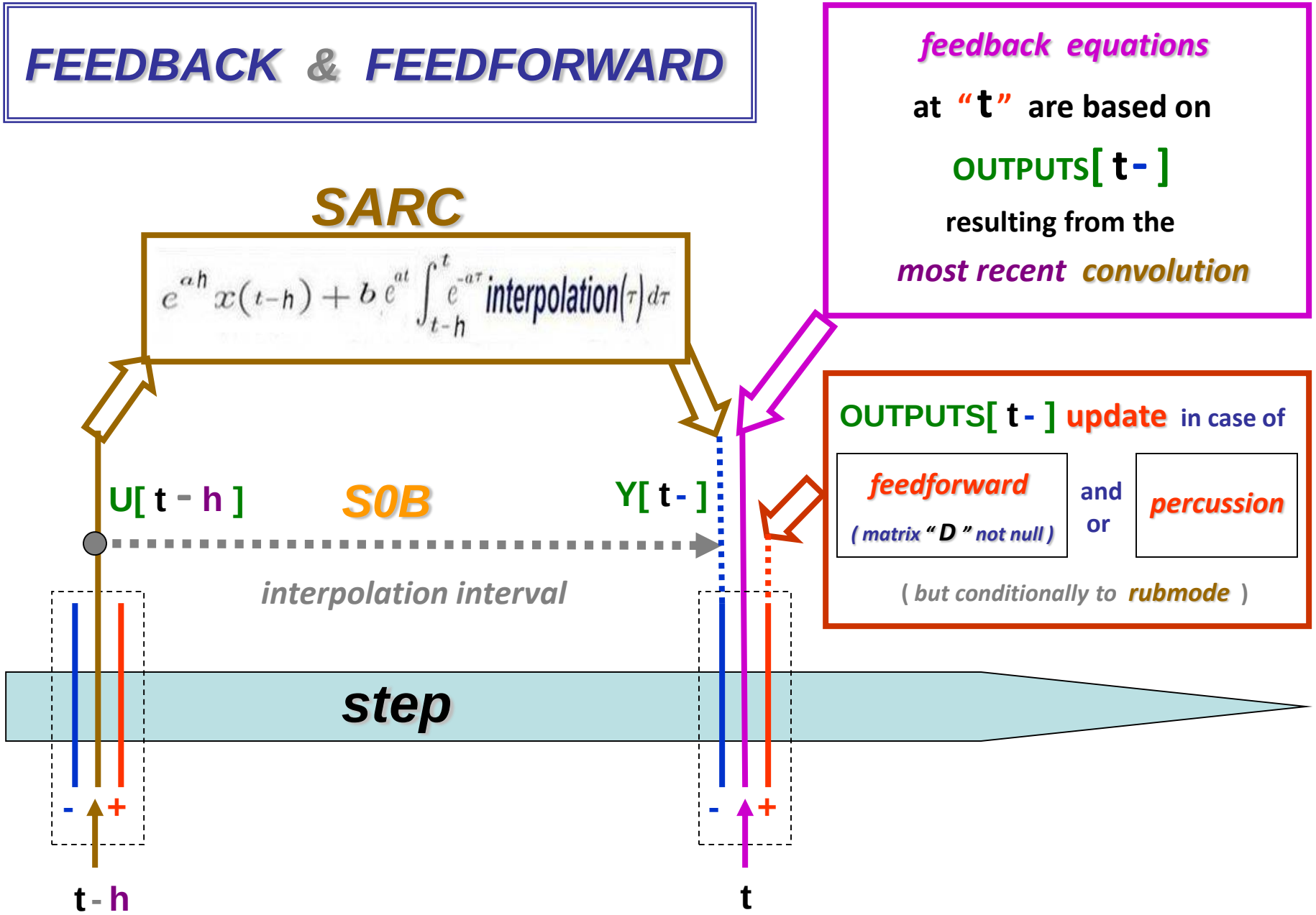





rubmode 0 t+ accurate (default)
1 t+ accurate except "AT" L&C percussion
2 t+ " immediate STEP changes " NEVER shown
This option does NOT change the accuracy of the NEXT steps
glitch < h
not shown
ideal SARC scheduling
step k
NAPA
Tplus
Tminus
NAPA
U k
[BOM] k
U k
ABCD k
SARC functions such as
POLES, ENERGY, PRINT, ...
can be called HERE
U k
ABCD k
U K+1
[BOM] K+1
h
X k+1 -
Y k+1 -
possible immediate information for percussion feedforward feedback
possible immediate information for percussion feedforward feedback
with coeff_XE k-1
U k-1 & U k
YE k-1
YM k-1
Δ t = 0
percuted X k -
recalculated Y
with percuted X k - and uptodate U k and CD k


ideal SARC scheduling
Tminus
NAPA
Tplus
U k-1
ABCD k-1
X k-1
Y k-1
U k
[BOM] k
U k
ABCD k
X̃ k-1
Ỹ k-1
possible
immediate
information
for percussion
feedforward
feedback
YE k-1
YM
computing end step k-1 = t-
values of
beginning
previous step
Y k-1
[BOM] k-1
regarding percussions
feedforward
feedback
NAPA takes decisions
with output values
one « h » old
(without Tminus step contribution)
current NAPA implementation
NAPA
Tminus
Tplus
U k
[BOM] k
U k-1
ABCD k-1
X k-1
Y k-1
U k
ABCD k
X̃ k-1
Ỹ k-1
YE k-1
YM k-1
computing end step k-1 = t-

FEEDBACK & FEEDFORWARD



feedback equations

at "**t**" are based on

OUTPUTS[t -]

resulting from the

most recent convolution

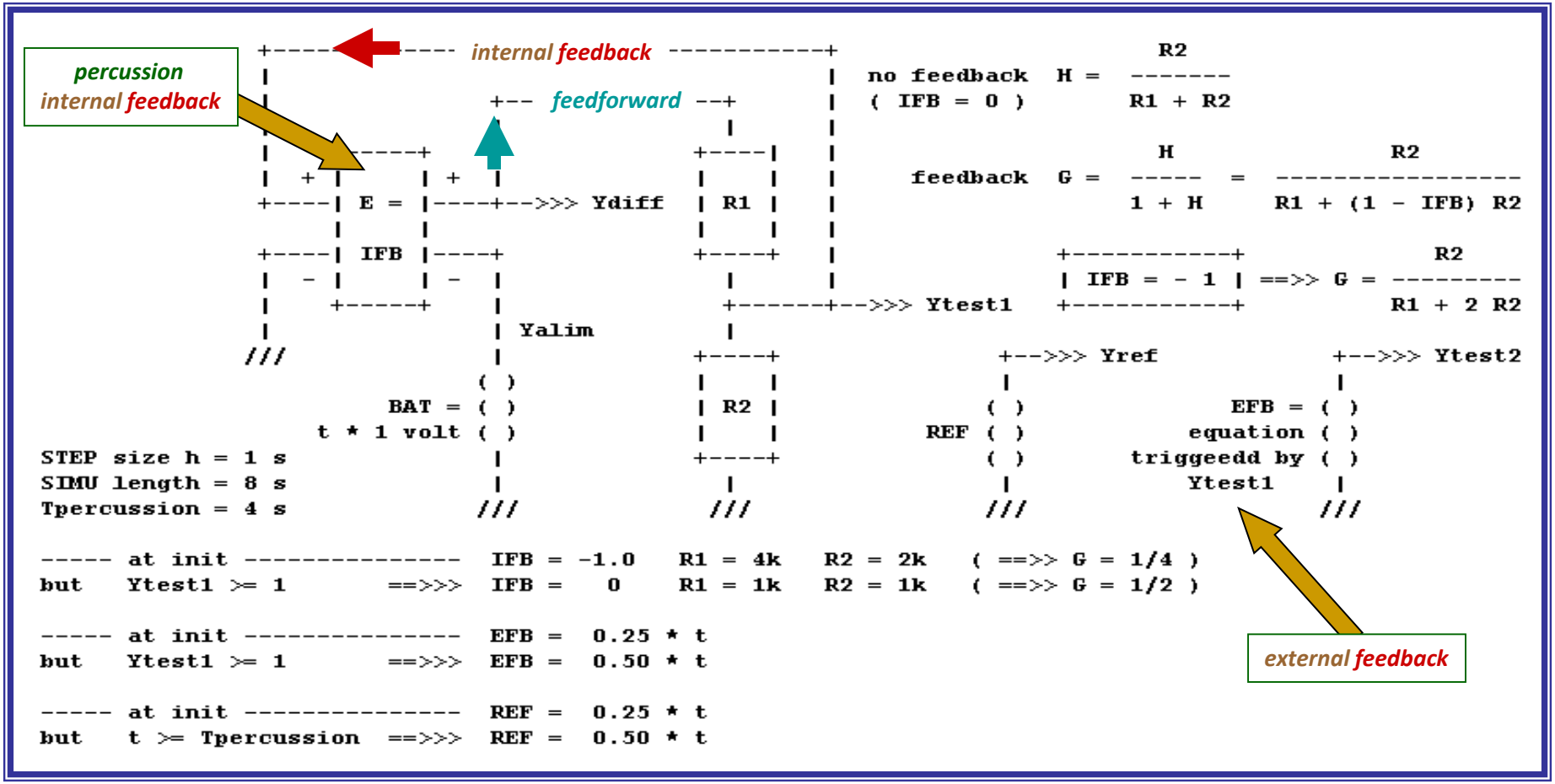
OUTPUTS[t -] update in case of

feedforward (matrix "**D**" not null)

and or

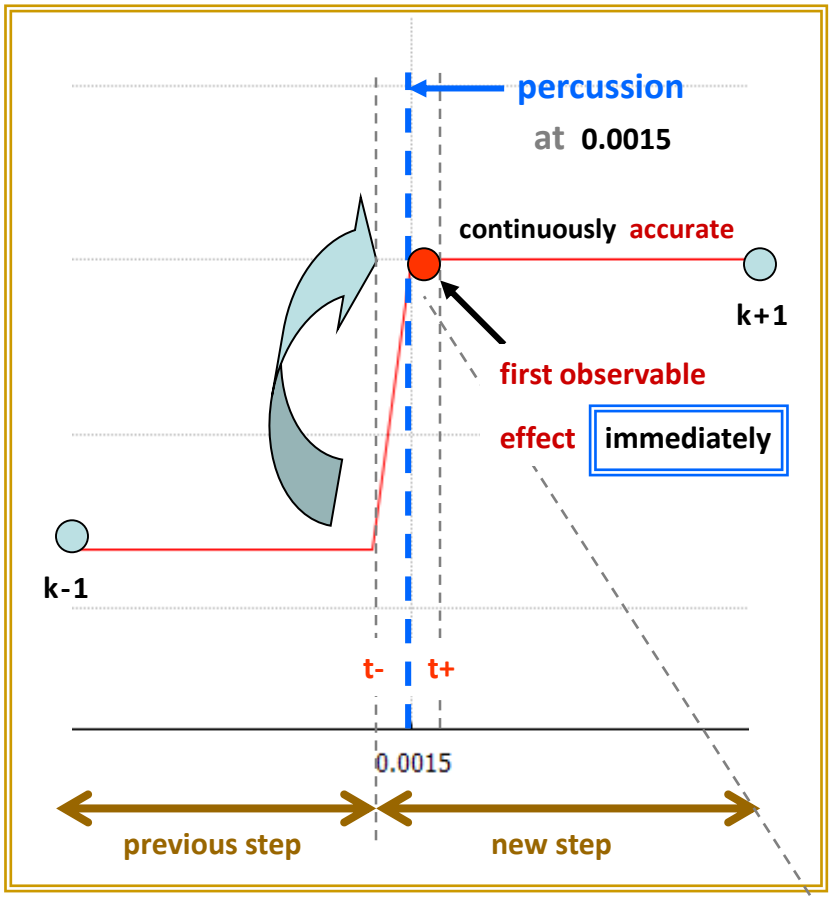
percussion

(but conditionally to *rubmode*)

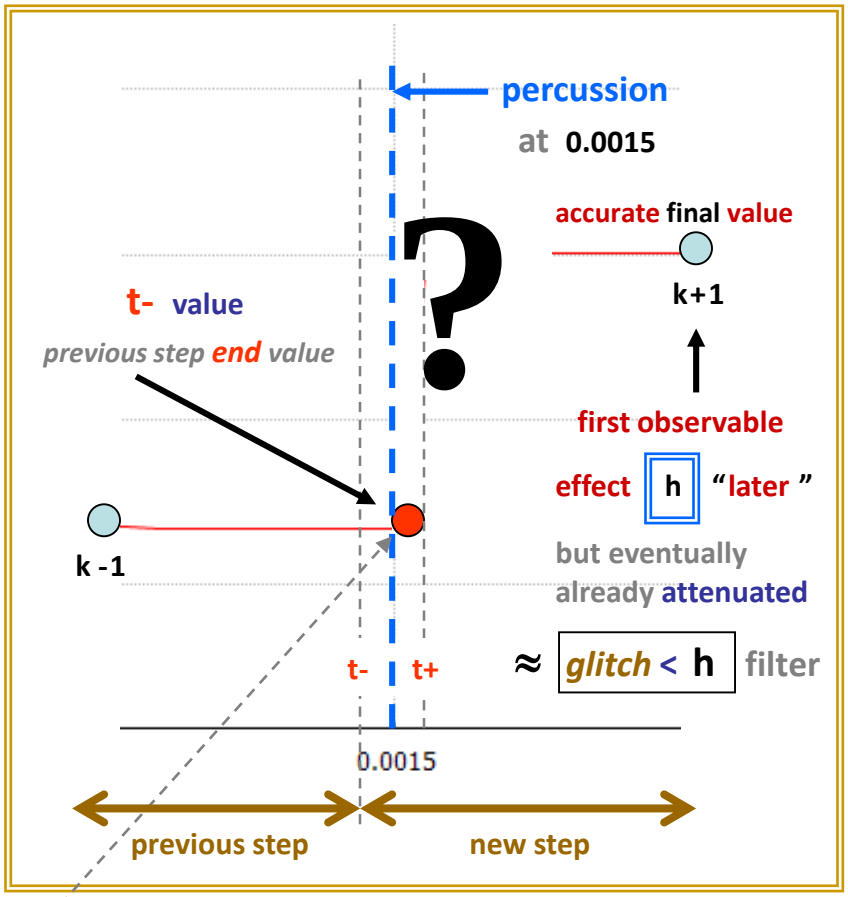



```
rubmode :  0  ;

NETLIST( /*-----------------------------------------*/
         [ V , BAT , Yalim  , Gnd                    ] ,
         [ E , IFB , Ydiff  , Yalim  , Ytest1 , Gnd ] ,
         [ R , R1  , Ydiff  , Ytest1                ] ,
         [ R , R2  , Ytest1 , Gnd                   ] ,
         /*-----------------------------------------*/
         [ V , EFB , Ytest2 , Gnd                   ] ,
         /*-----------------------------------------*/
         [ V , REF , Yref   , Gnd                   ]
         /*-----------------------------------------*/ ) ;

OUTPUTS( V(Yalim) , V(Ydiff) , V(Ytest1) , V(Ytest2) , V(Yref) ) ;
```

rubmode = 0 *(default)*

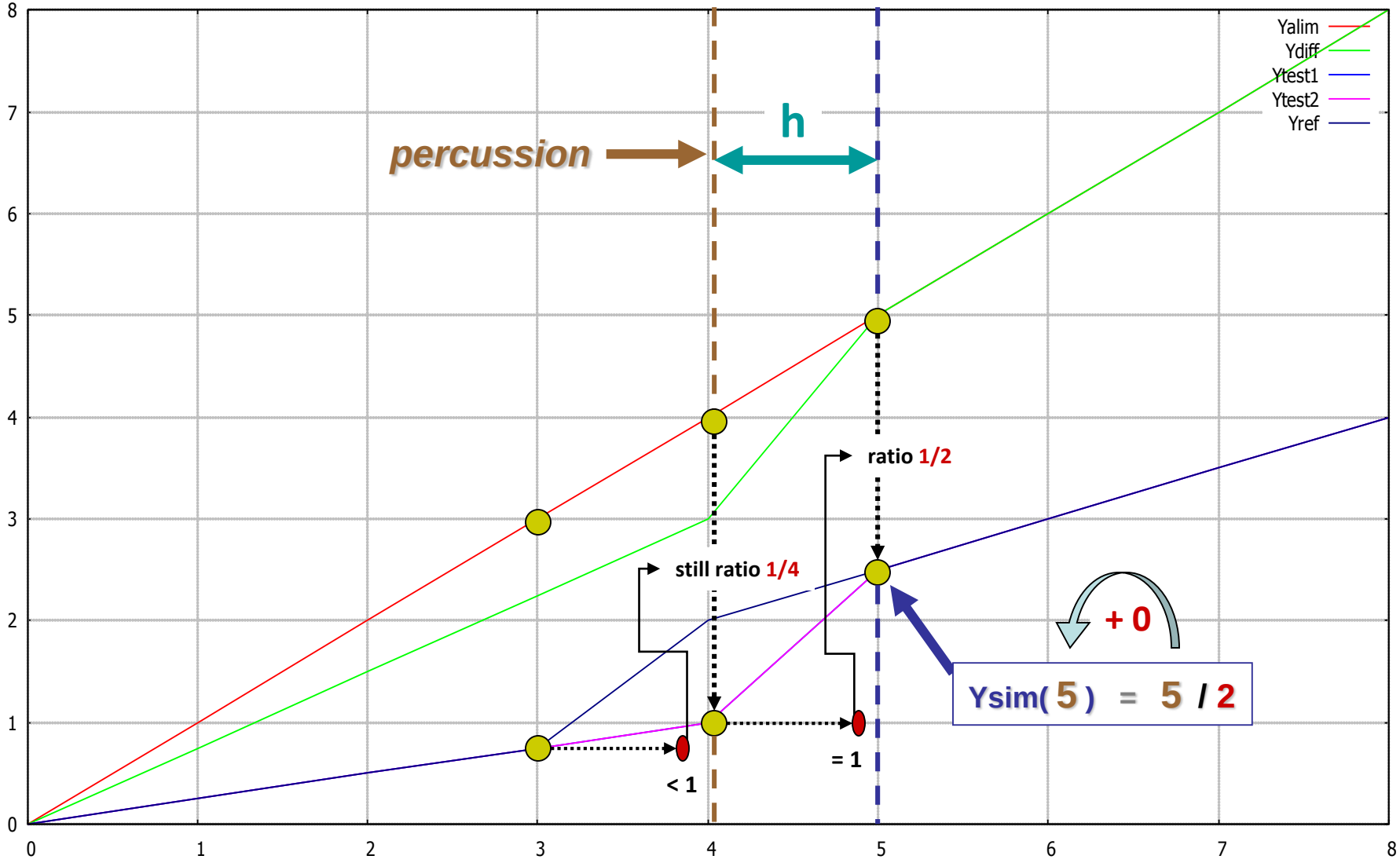


rubmode ≠ 0

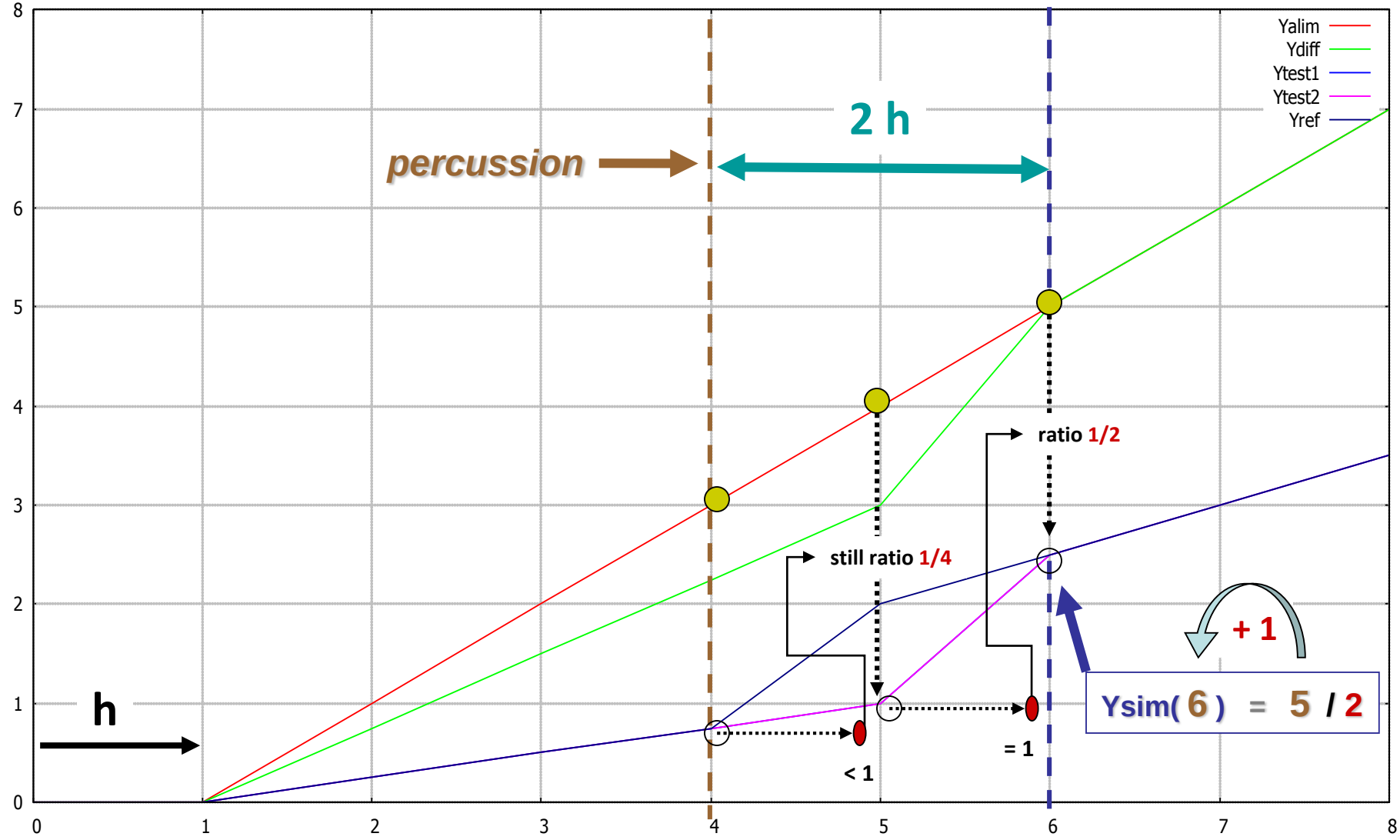


warning: in ***rubmode*** *(when no explicit NAPA delay)* the overall simulation waveform appears just ***shifted in time***

(*rubmode* = *0*)


Yalim
Ydiff
Ytest1
Ytest2
Yref
percussion
h
ratio 1/2
still ratio 1/4
+ 0
Ysim(5) = 5 / 2
< 1
= 1
0
1
2
3
4
5
6
7
8

(*rubmode* = 2)


Yalim
Ydiff
Ytest1
Ytest2
Yref
2 h
percussion
h
ratio 1/2
still ratio 1/4
< 1
= 1
+ 1
Ysim(6) = 5 / 2