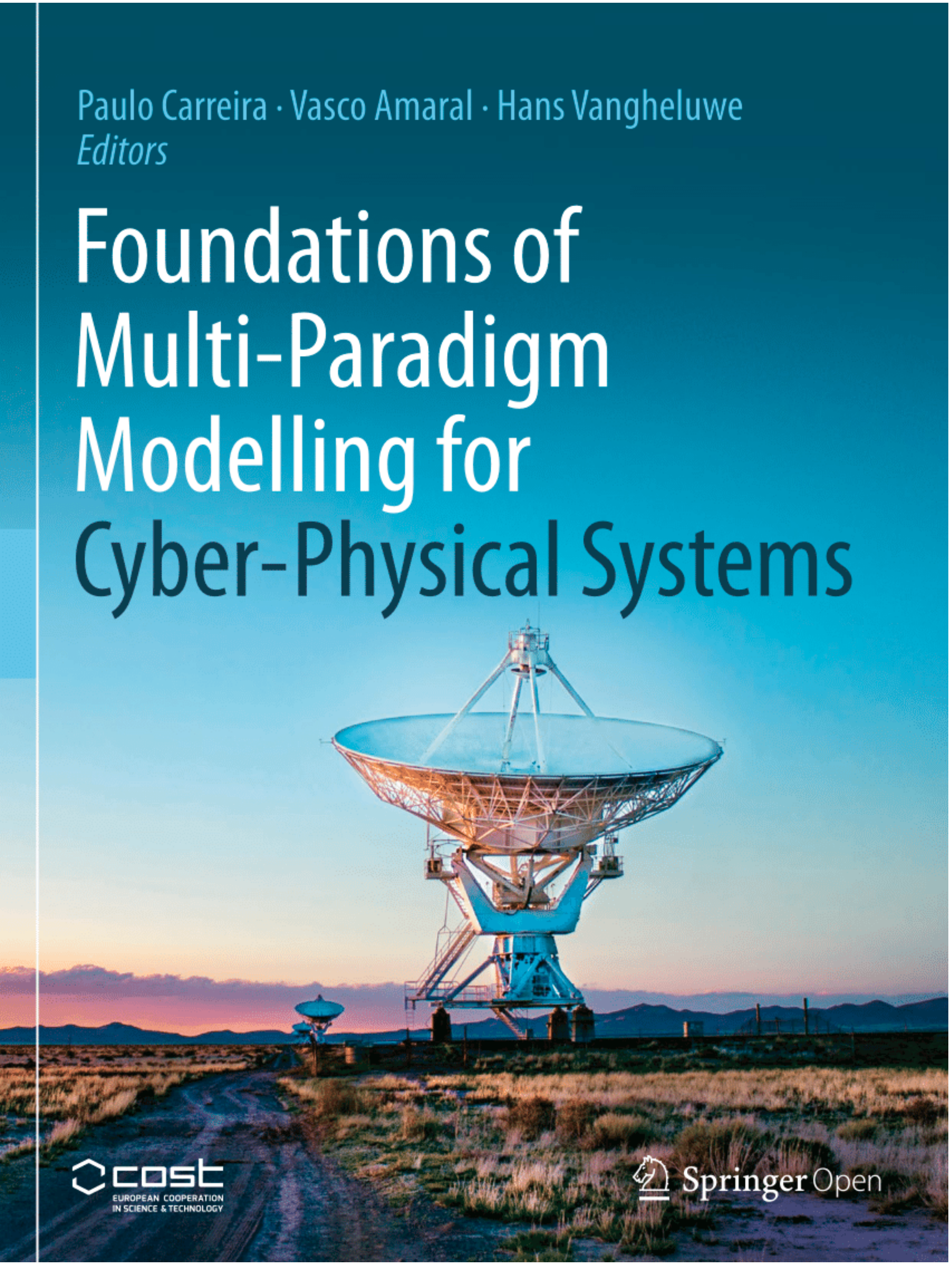
Causal Block Diagrams

Hans Vangheluwe

Universiteit Antwerpen
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp
Nexor
Cyber-Physical Systems
University of Antwerp
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK
MPM4CPS
cost

Physical Systems Modelling

- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure



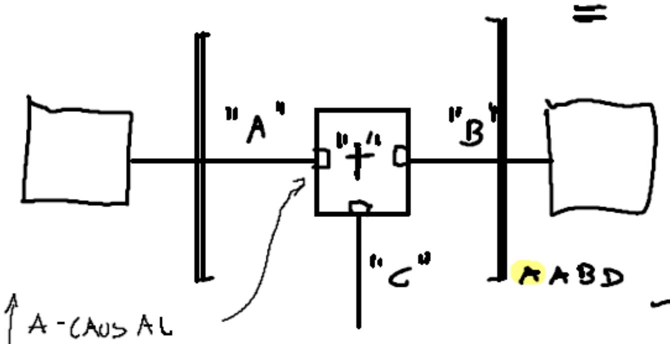
Gomes C., Denil J., Vangheluwe H. (2020) Causal-Block Diagrams: A Family of Languages for Causal Modelling of Cyber-Physical Systems. In: Carreira P., Amaral V., Vangheluwe H. (eds) Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems. Springer, Cham.
https://doi.org/10.1007/978-3-030-43946-0_4

CBD — CAUSAL BLOCK DIAGRAM

ABD — A-CAUSAL BLOCK DIAGRAM

"COMPUTATIONAL CAUSALITY"

CAUSE ⟶ CONSEQUENCE



=

$$mvA + mvB + mvC = \phi$$

$$(mvA, mvB, mvC) \in \mathbb{R}^3$$

A-CAUSAL

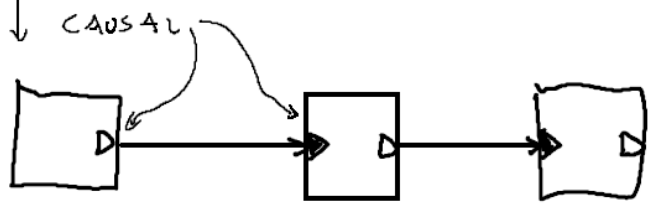
$$+ : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$$

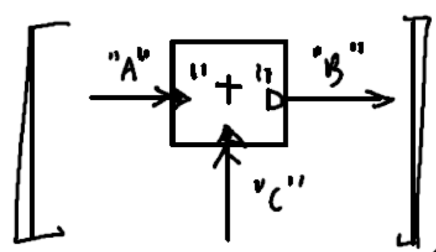
MATH

"WHAT"

CAUSALITY ASSIGNMENT

CAUSAL





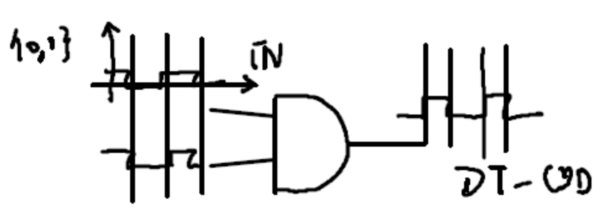
=

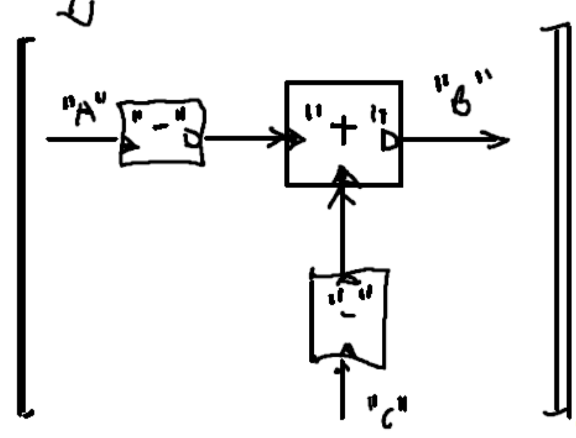
vA, vB, vC : float

vB := vA + vC

CODE

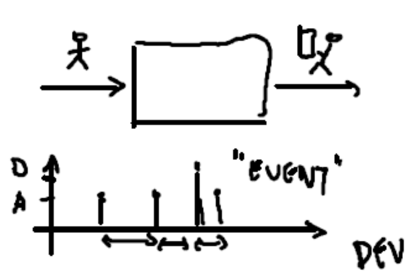
"HOW"

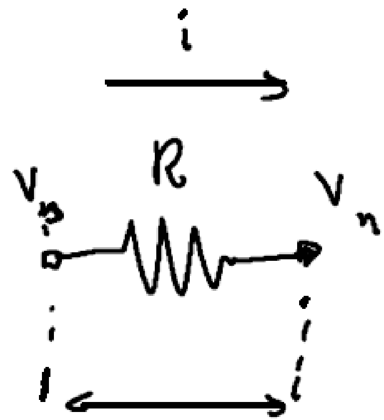




=

vB := - vA - vC



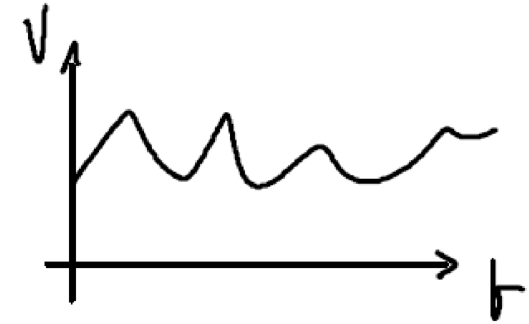


$$V = v_p - v_n$$

$$V := R \times i$$

$$R := V/i$$

$$i := V/R$$

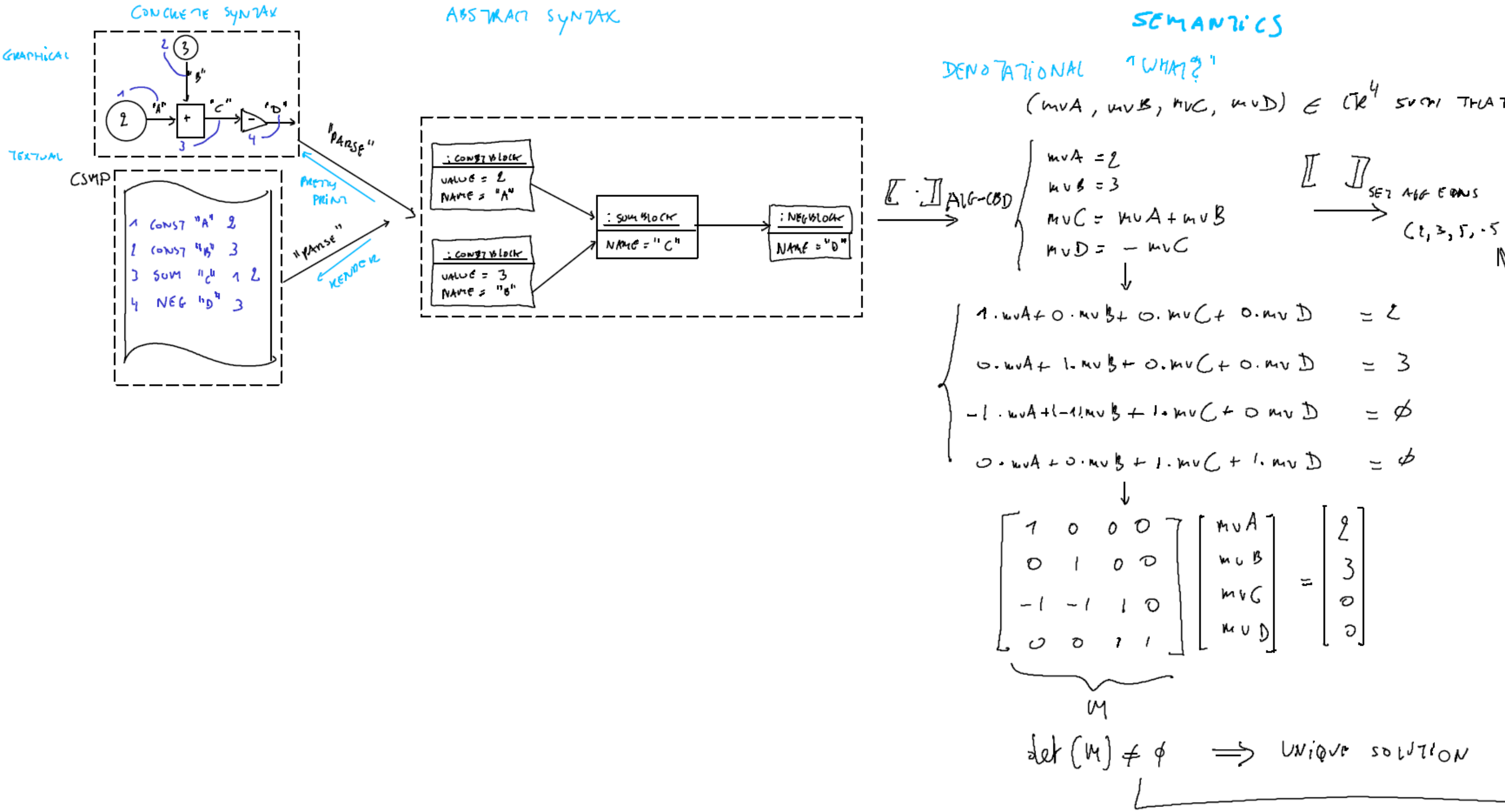


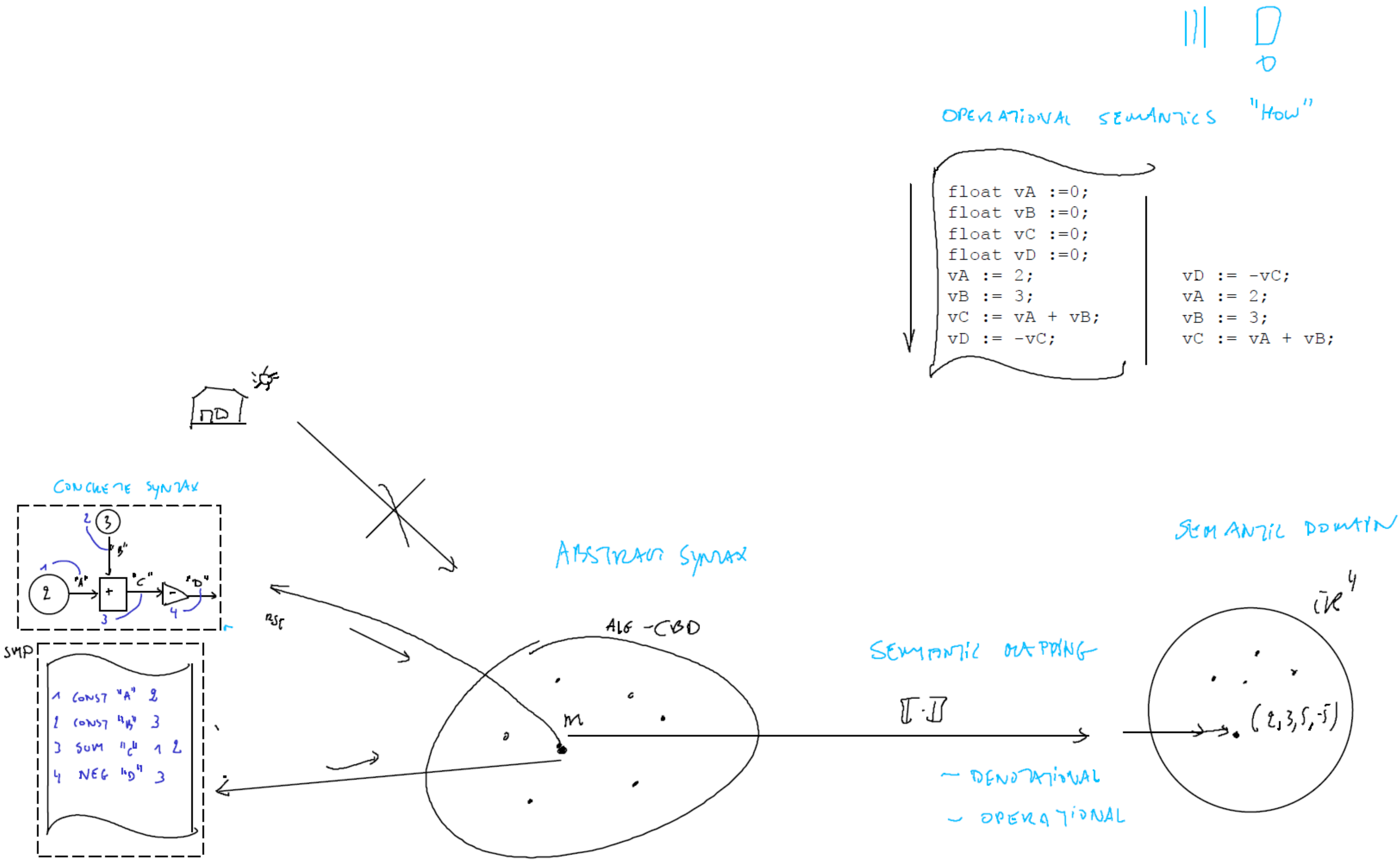
$$V - Ri = \phi$$

↑ ↑

MODELICA ⇝ SIMULINK

CAUSALITY ASGN.

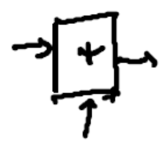
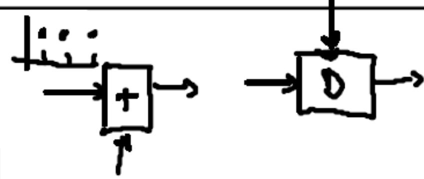
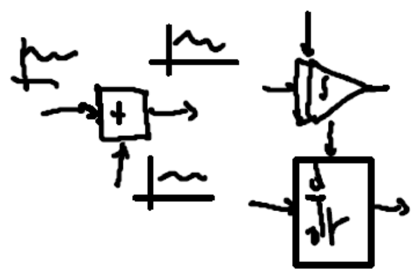

CONCRETE SYNTAX
ABSTRACT SYNTAX
SEMANTICS
GRAPHICAL
TEXTUAL
CSMP
1 CONST "A" 2
2 CONST "B" 3
3 SUM "C" 1 2
4 NEG "D" 3
"PARSE"
PRETTY PRINT
"PARSE"
RENDER
: CONSTBLOCK
VALUE = 2
NAME = "A"
: CONSTBLOCK
VALUE = 3
NAME = "B"
: SUMBLOCK
NAME = "C"
: NEGBLOCK
NAME = "D"
DENOTATIONAL "WHAT?"
(mvA, mvB, mvC, mvD) ∈ ℝ^4 SUCH THAT
⟦ · ⟧ ALG-EQ
mvA = 2
mvB = 3
mvC = mvA + mvB
mvD = − mvC
1·mvA + 0·mvB + 0·mvC + 0·mvD = 2
0·mvA + 1·mvB + 0·mvC + 0·mvD = 3
−1·mvA + (−1)·mvB + 1·mvC + 0·mvD = 0
0·mvA + 0·mvB + 1·mvC + 1·mvD = 0
⟦ ⟧ SET ALG EQNS
(2, 3, 5, −5)
M
det(M) ≠ 0 ⟹ UNIQUE SOLUTION

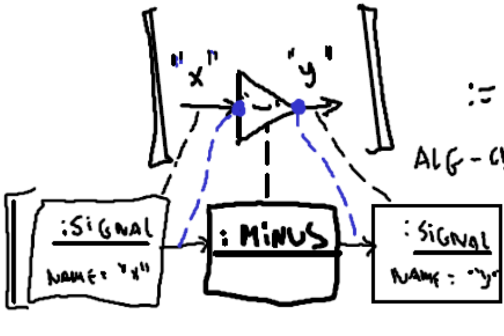

OPERATIONAL SEMANTICS
"HOW"
float vA :=0;
float vB :=0;
float vC :=0;
float vD :=0;
vA := 2;
vB := 3;
vC := vA + vB;
vD := -vC;
vD := -vC;
vA := 2;
vB := 3;
vC := vA + vB;
CONCRETE SYNTAX
"A"
"B"
"C"
"D"
ABSTRACT SYNTAX
ALG-CBD
SEMANTIC DOMAIN
SEMANTIC MAPPING
1 CONST "A" 2
2 CONST "B" 3
3 SUM "C" 1 2
4 NEG "D" 3
- DENOTATIONAL
- OPERATIONAL

HIERARCHY

FLATTEN

SEMANTICS

TIME ↓	FLAT CBD	SYNTAX	DENOTATIONAL "WHAT"	OPERATIONAL "HOW"
{NOW}	ALGEBRAIC (ALG-CBD)	 NO LOOPS / WITH LOOPS		
ℕ	DISCRETE-TIME (DT-CBD)			
ℝ	CONTINUOUS-TIME (CT-CBD)			



ALG-CBD

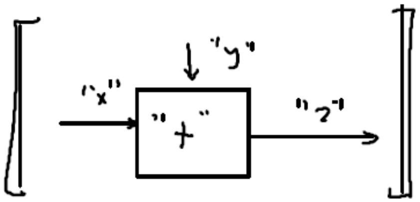
ALG-CBD

$:=$ $[\![\text{"y"}]\!] = -[\![\text{"x"}]\!]$

$mvy := -\, mvx$

$mvx, mvy \in \mathbb{R}$

$- : \mathbb{R} \to \mathbb{R}$



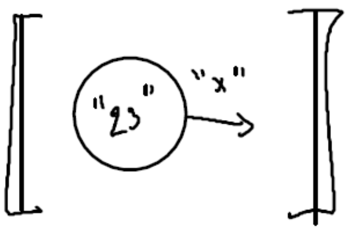
ALG-CBD

$:=$ $[\![\text{"z"}]\!] = [\![\text{"x"}]\!] + [\![\text{"y"}]\!]$

$mvz = mvx + mvy$

$mvx, mvy, mvz \in \mathbb{R}$

$+ : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$

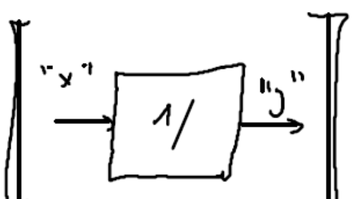


ALG-CBD

$:=$ $[\![\text{"x"}]\!] = [\![\text{"23"}]\!]$

$mvx = 23$

$mvx \in \mathbb{R}$



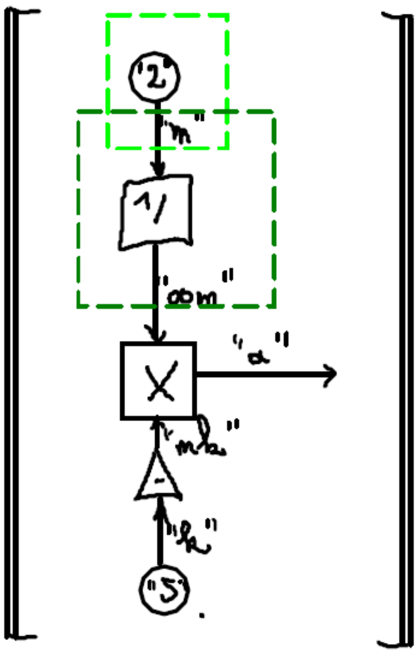
ALG-CBD

$:=$ $[\![\text{"y"}]\!] = 1/[\![\text{"x"}]\!]$

$mvy = 1/mvx$

$mvx \in \mathbb{R}\setminus\{0\},\ mvy \in \mathbb{R}$

$mvx, mvy \in \mathbb{R} \cup \{\text{EXCEPT}\}$



ALG-CHD

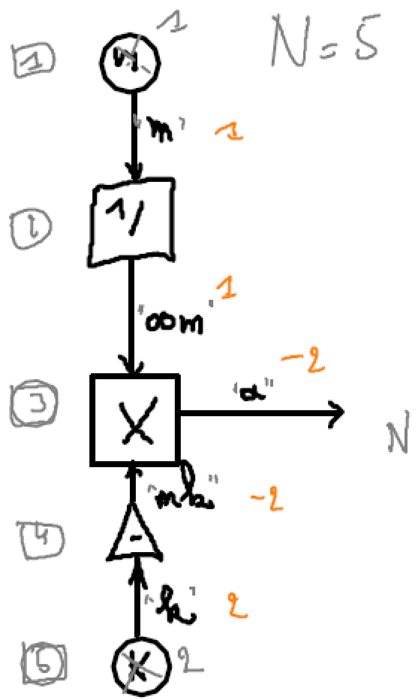
:=

mvm, mvoom, mvm, mva, mvmk, mvk ∈ ℝ

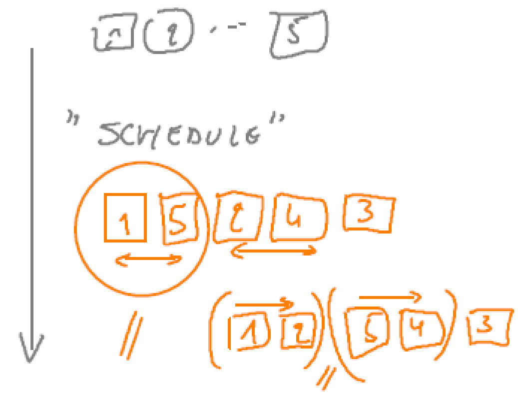
SUCH THAT mvm ≠ ∅

{ mvm = 2
mvoom = 1/mvm
mva = mvoom × mvmk
mvmk = -mvk
mvk = 5 }

= (2, 0.5, 2.5, -5, 5) ∈ $\mathbb{R}^5$



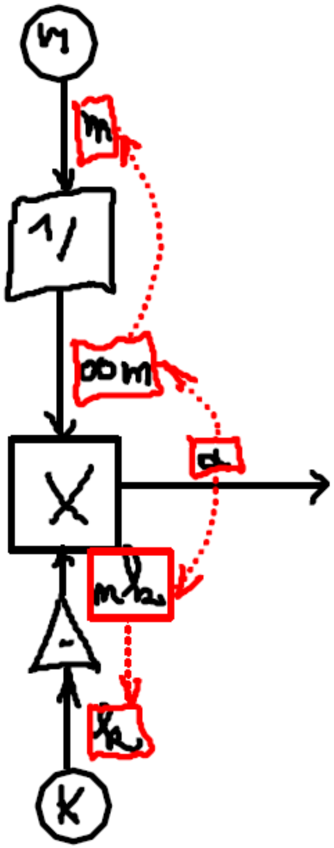
	vm	voom	va	vmk	vk
	UK	UK	UK	UK	UK
[1]	1	UK	UK	UK	UK
[2]	1	1	UK	UK	UK
[3]	1	1	UK	UK	UK
[4]	1	1	UK	UK	UK
[5]	1	1	UK	UK	2
[1]	1	1	UK	UK	2
[2]	1	1	UK	UK	2
N [3]	1	1	UK	UK	2
[4]	1	1	UK	-2	2
[5]	1	1	UK	-2	2
[1]	1	1	UK	-2	2
[2]	1	1	UK	-2	2
N [3]	1	1	-2	-2	2
[4]					
[5]					



WORST-CASE

# ITER	$\Theta(N)$	$\Theta(1)$
# COMP	$\Theta(N^2)$	$\Theta(N)$

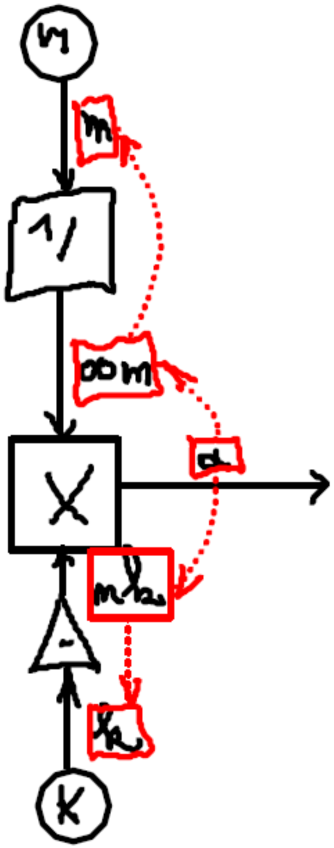
DISTRIBUTED



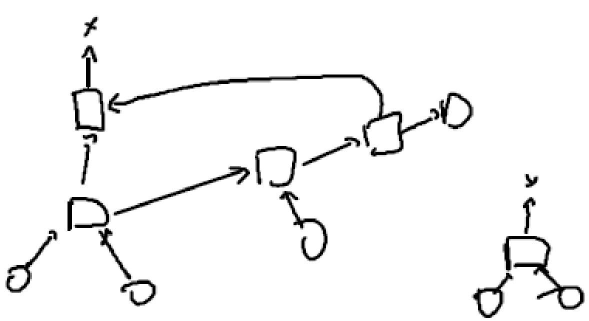
DEPENDENCY GRAPH

schedule = [m, oom, k, mk, a]

DEPENDENCY GRAPH



schedule = [m, oom, k, mk, a]



operational semantics

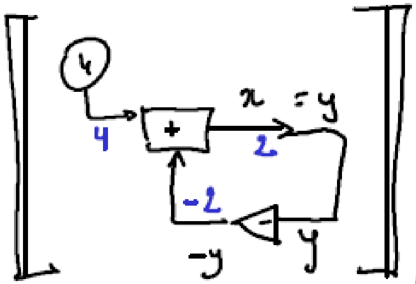
$\Theta(N)$

```
depGraph = buildDepGraph(CBD)
schedule = topologicalSort(depGraph)
```

$\Theta(N)$

```
for block in schedule:
  block.compute()
```

$\Theta(N)$



:=

$$\begin{cases} x + y = 4 \\ x - y = \emptyset \end{cases} \qquad \begin{aligned} x &= 4 - y \\ y &= x \end{aligned}$$

ALG-COD

SET LINEAR EQNS.

2 UNKNOWNS

2 EQNS

$$\begin{cases} 1 \times x + 1 \times y = 4 \\ 1 \times x - 1 \times y = \emptyset \end{cases}$$

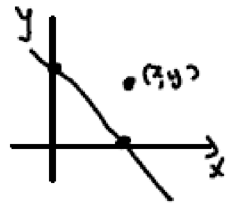
$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

$$\det \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = -2 \neq \emptyset$$

$$x = \frac{\begin{vmatrix} 4 & 1 \\ \emptyset & -1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}} = \frac{-4}{-2} = 2$$

$$y = \frac{\begin{vmatrix} 1 & 4 \\ 1 & \emptyset \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}} = \frac{-4}{-2} = 2$$

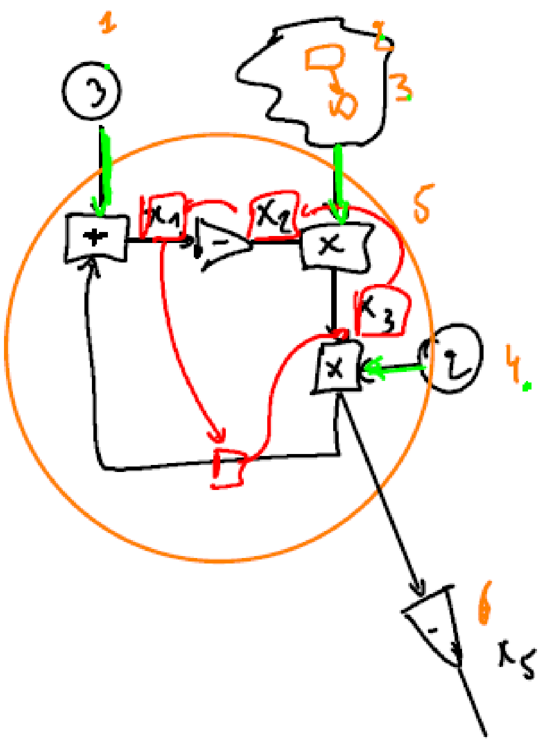
$$\begin{cases} 2x + 2y = 4 \\ x + y = 2 \end{cases}$$



$$y = 2 - x$$

$$\begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$\begin{vmatrix} 2 & 2 \\ 1 & 1 \end{vmatrix} = \phi$$



$$\begin{cases} k_{11} x_1 + k_{21} x_2 + \dots k_{n1} x_n = K_1 \\ k_{12} x_1 + k_{22} x_2 + \dots k_{n2} x_n = K_2 \\ \vdots \\ k_{1n} x_1 + k_{2n} x_2 + \dots k_{nn} x_n = K_n \end{cases}$$

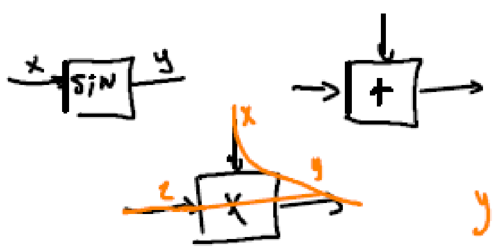
$$\underbrace{\begin{bmatrix} k_{11} & \dots \\ \vdots & \\ & k_{nn} \end{bmatrix}}_{K} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \underbrace{\begin{bmatrix} K_1 \\ \vdots \\ K_n \end{bmatrix}}_{K}$$

UNIQUE? $\det(K) \neq \emptyset$

$(x_1, \dots, x_n)$

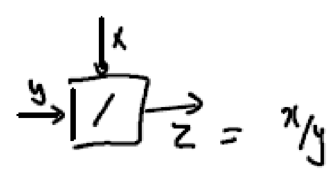
GAUSSIAN ELIM $O(n^3)$

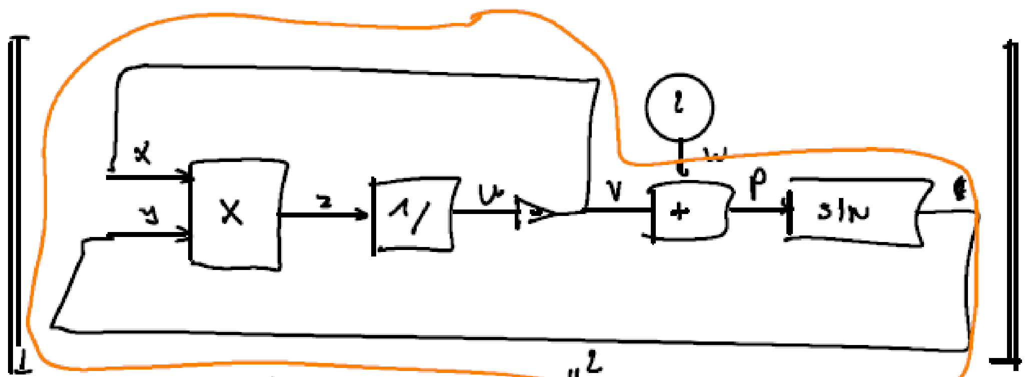
BLOCKS IN "LINEAR LOOP"



$y = x \times z$

$xy - x \times z = \emptyset$





$(x, y, z, u, v, w, p) \in \mathbb{R}$

UNIQUE ?

COMPLEXITY ? $O(..)$

$x \times y - z = 0 \quad \varepsilon_1$

$z - 1/u = 0 \quad \varepsilon_2$

$\vdots$

$\to \varphi$

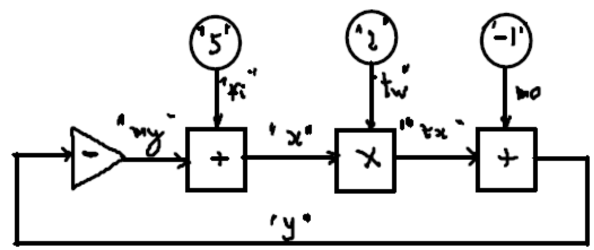
$(x_0, y_0, z_0, u_0, v_0, w_0, p_0)$

$(x_1, y_1, z_1, u_1, v_1, w_1, p_1)$

"algebraic loop"

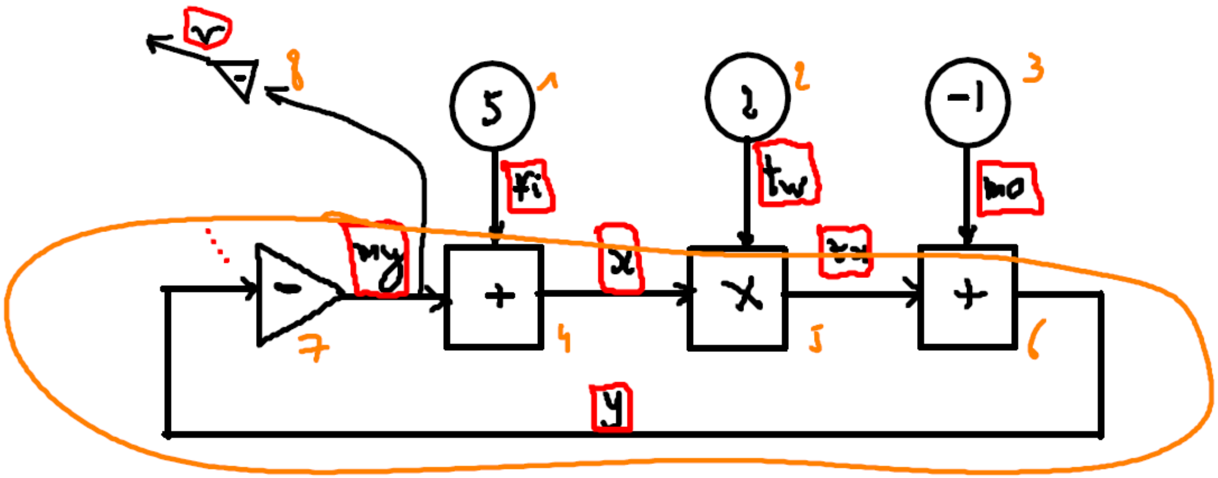
$$mvfi, mvx, mvtw, mvtx, mvmo, mvy, mvny \in \mathbb{R}$$

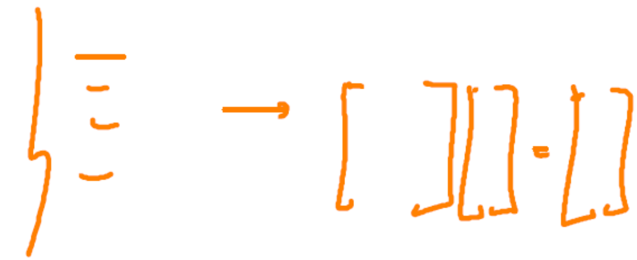
such that...

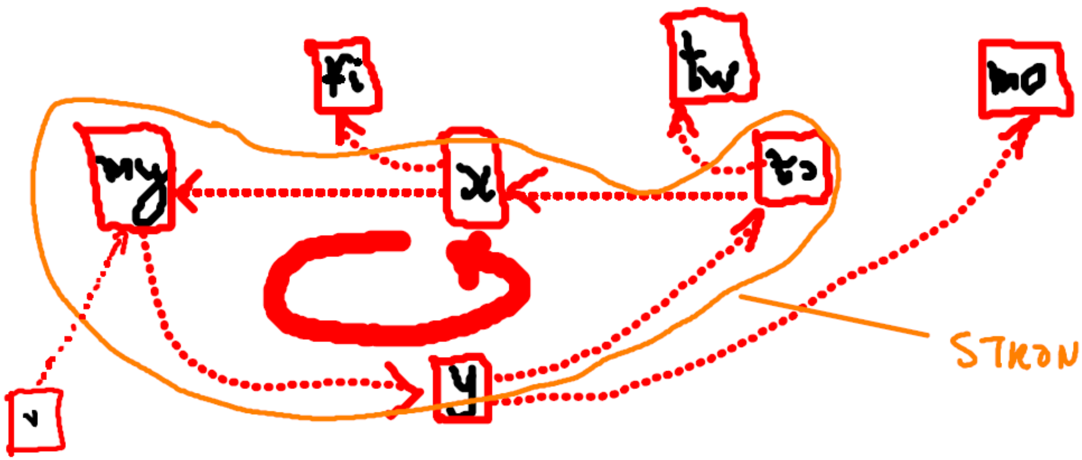
$\Big[\!\Big[$ 

$\Big]\!\Big]_{ALG\text{-}CBD}$:=

$$\begin{cases} mvfi = 5 \\ mvx = mvfi + mvny \\ mvtw = 2 \\ mvtx = mvtw \times mvx \\ mvmo = -1 \\ mvy = mvmo + mvx \\ mvny = -mvy \end{cases} = (5, 2, 2, 4, -1, 3, -3) \in \mathbb{R}^7$$







STRONG COMPONENT

[{1},{2},{3}, {4,5,6,7}, {8}]

TARJAN 1974

$\Theta(N+E)$

operational semantics

```
schedule = topologicalSortAndLoopDetect(depGraph(CBD))

for genBlock in schedule:
  genBlock.compute()
```

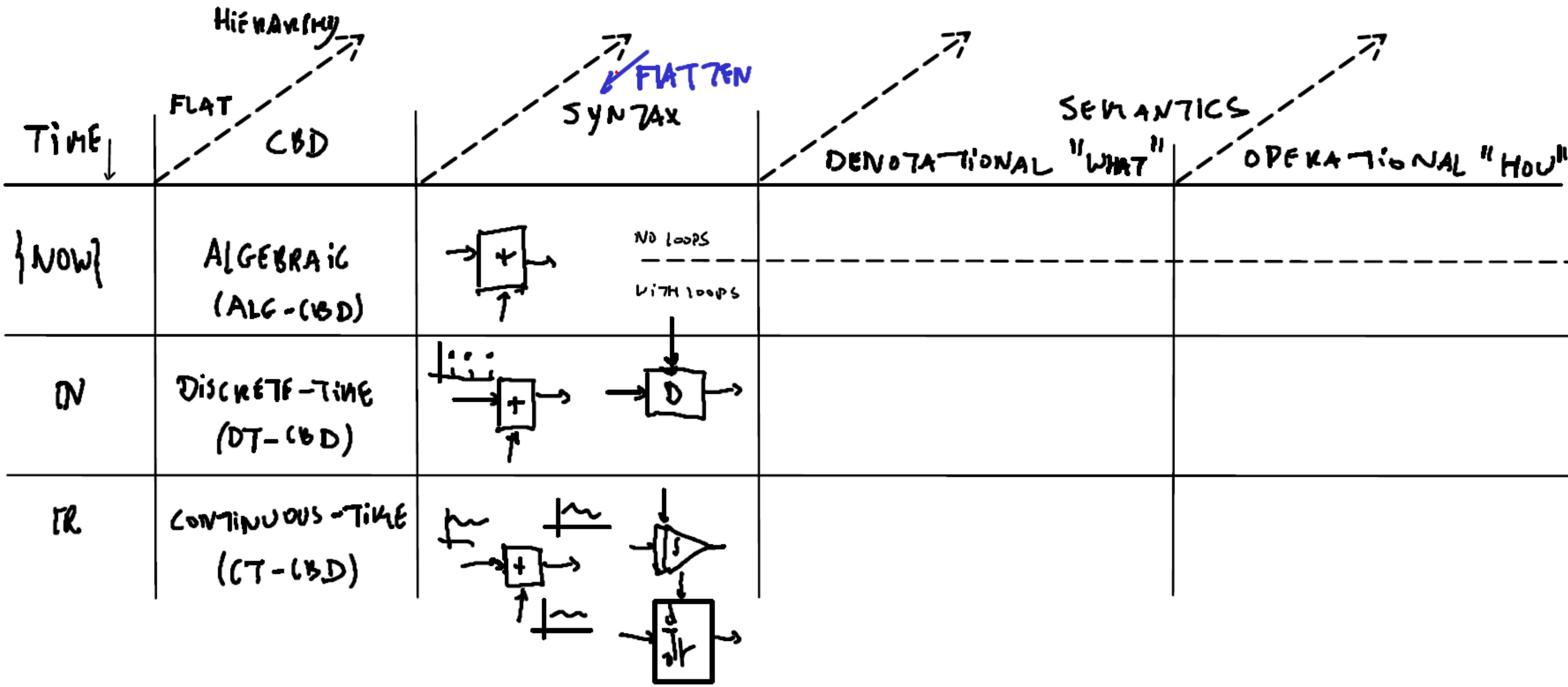
$\Theta(N+E)$

$\Theta(N)$?

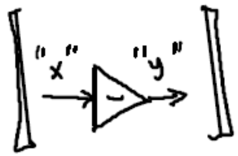
some sets of coupled eqns (size m)

solver complexity $\leq \Theta(m^3)$

usually $m \ll N$


HIERARCHY
FLATTEN
FLAT
CBD
TIME
SYNTAX
SEMANTICS
DENOTATIONAL "WHAT"
OPERATIONAL "HOW"
{NOW}
ALGEBRAIC
(ALG-CBD)
NO LOOPS
WITH LOOPS
ℕ
DISCRETE-TIME
(DT-CBD)
ℝ
CONTINUOUS-TIME
(CT-CBD)
+
D

DISCRETE-TIME CBD (DT-CBD)



$:= [\![y]\!] = -[\![x]\!]$ $\quad m_y(i) = -m_x(i), \ \forall i \in \mathbb{N}_0$

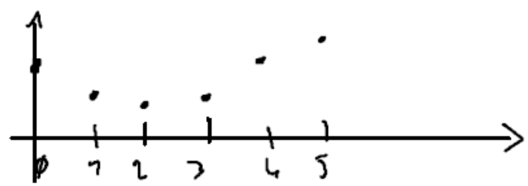
$m_x, m_y \in \mathbb{N}_0 \to \mathbb{R}$

$- : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

x_i

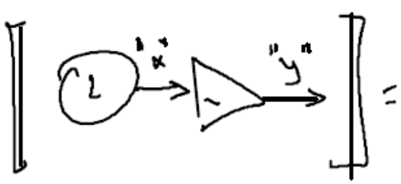
$x(i)$ $\quad i \in \mathbb{N}_0$

mx

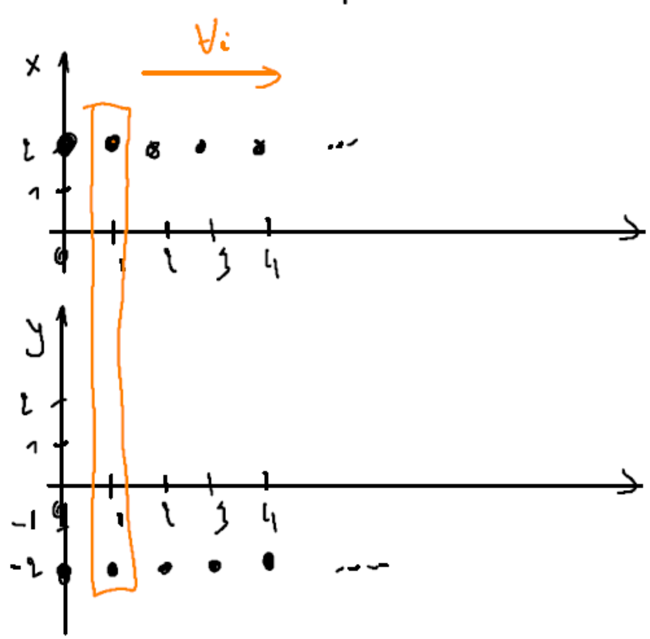


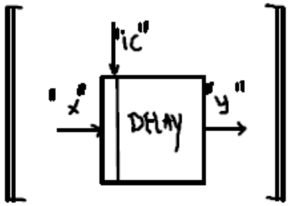
$m_x = \{ 0 \to 10.3, \ 1 \to 5.5, \ 2 \to 3.33, \ldots \}$

$[(0, 10.3), (1, 5.5), (2, 3.33), \ldots]$



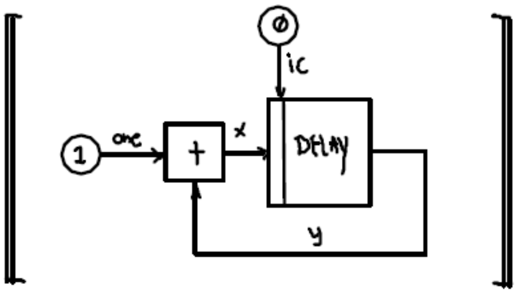
$=$ $\begin{cases} x(i) = 2 \\ y(i) = -x(i) \end{cases} \quad \forall i \in \mathbb{N}_0$





:= $\begin{cases} m_y(i) = m_x(i-1), & \forall i \in \mathbb{N} \\ m_y(\phi) = m_{ic}(\phi) \end{cases}$

$v_{IC}, v_y, v_x, v_{one} \in \mathbb{N}_\phi \to \mathbb{R}$

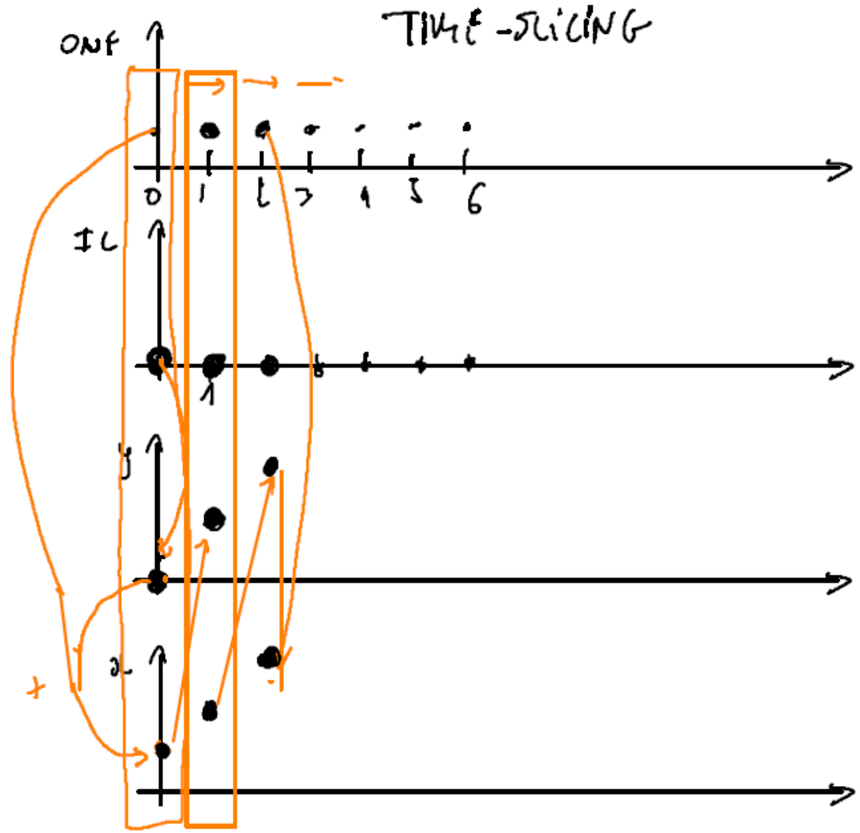


:= $\begin{cases} v_{IC}(i) = \phi, \ \forall i \in \mathbb{N}_\phi \\ v_y(i) = v_x(i-1), \ \forall i \in \mathbb{N} \\ v_y(\phi) = ic(\phi) \\ v_x(i) = v_{one}(i) + v_y(i), \ \forall i \in \mathbb{N}_\phi \\ v_{one}(i) = 1, \ \forall i \in \mathbb{N}_\phi \end{cases}$ $= ([0, 0, 0, \ldots], [0, 1, 1, \ldots], [1, 2, 3, \ldots], [1, 1, 1, \ldots]) \in \mathbb{R}$

↓ ALG

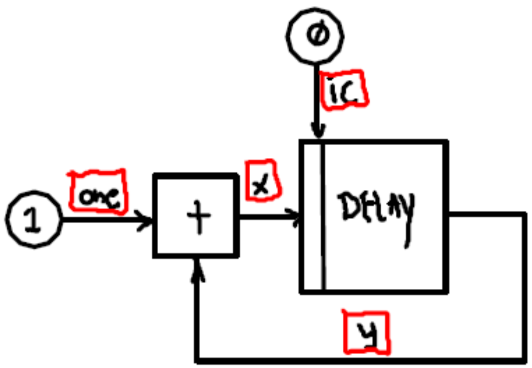
COUNTER

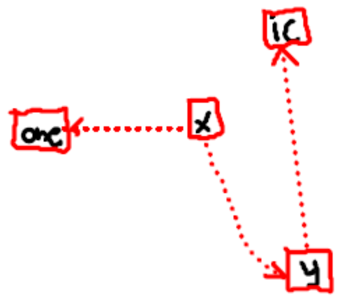
TIME-SLICING



operational semantics

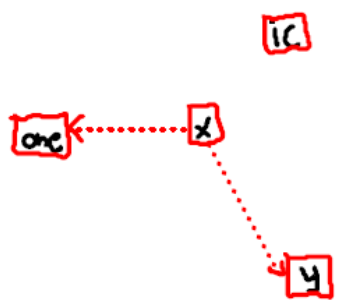
```
i = 0

while (not end_condition(i, ...)):

  depGraph = buildDepGraph(CBD)
  schedule = loopDetectAndTopSort(depGraph)

  for gblock in schedule:
    gblock.compute()

  i++
```





$i = \phi$

schedule = [one, IC, y, x]



$\forall i \in \mathbb{N}$

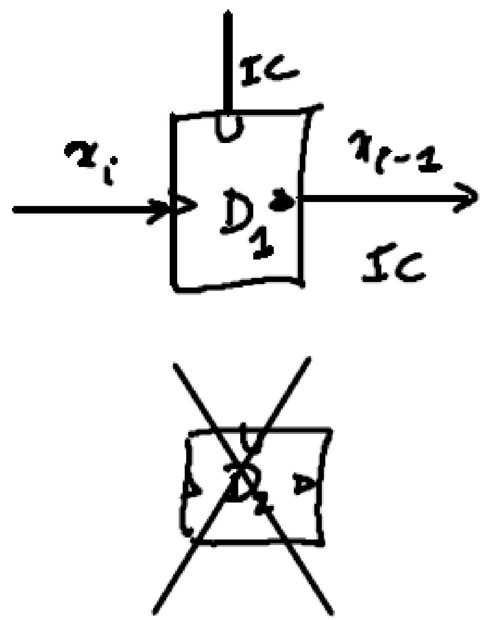
schedule = [y, one, x, IC]

operational semantics

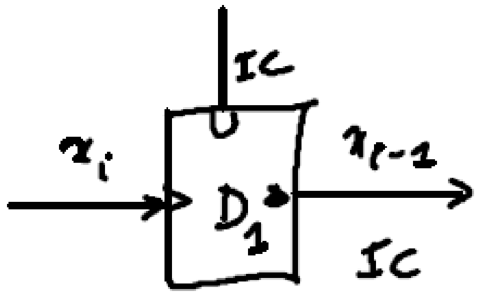
```
i = 0

while (not end_condition(i, ...)):

  depGraph = buildDepGraph(CBD)
  schedule = loopDetectAndTopSort(depGraph)

  for gblock in schedule:
    gblock.compute()

  i++
```

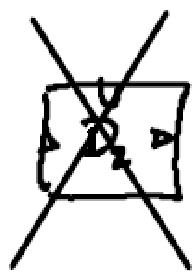
```
i = 0

if (not end_condition(i, ...)):
  depGraph = buildDepGraph(CBD)
  schedule = loopDetectAndTopSort(depGraph)

  for gblock in schedule:
    gblock.compute()
else:
  exit()

i = 1

while (not end_condition(i, ...)):

  depGraph = buildDepGraph(CBD)
  schedule = loopDetectAndTopSort(depGraph)

  for gblock in schedule:
    gblock.compute()

  i++
```

memory over more than 1 time-slice



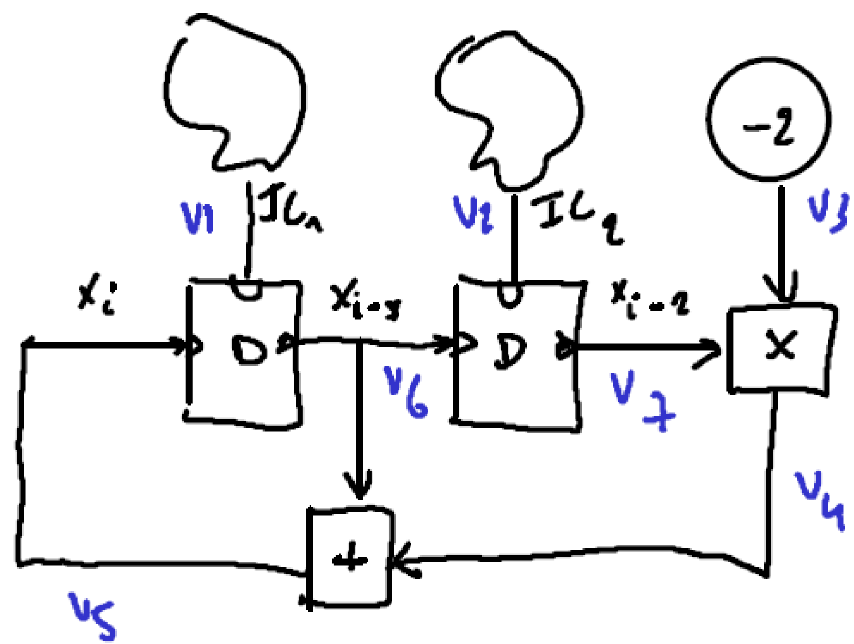
Memory over more than 1 time-slice

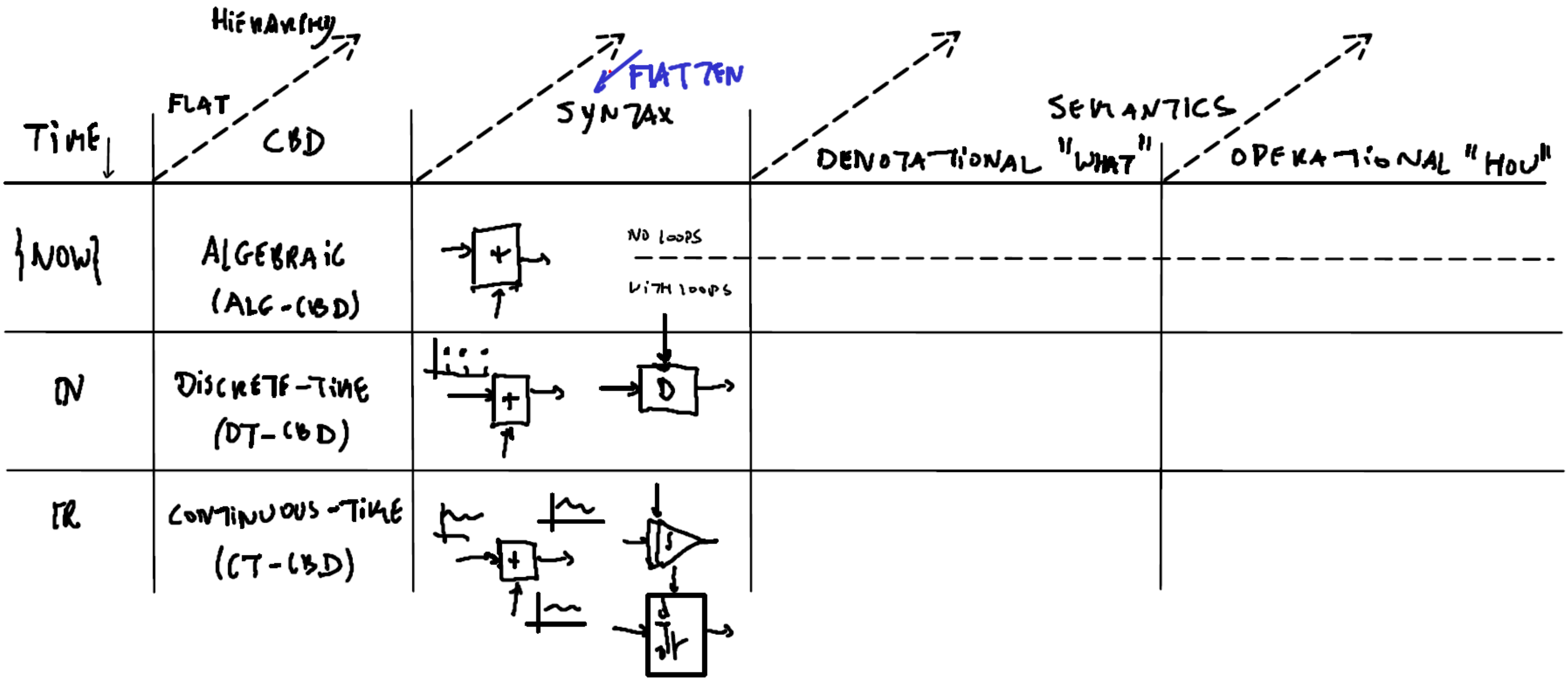




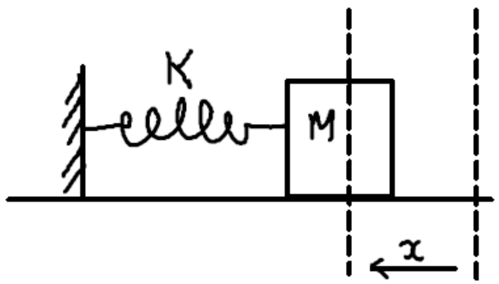
$$x_{i+1} = x_i - 2 \times x_{i-1}$$

$$x_i = x_{i-1} - 2 \times x_{i-2}$$

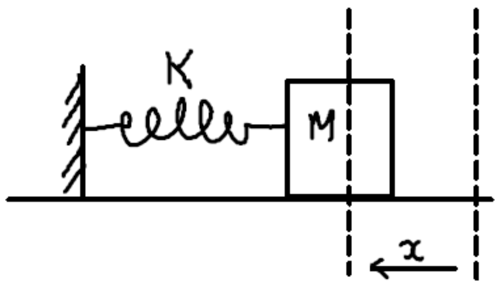




HIERARCHY
FLAT
CBD
SYNTAX
FLATTEN
SEMANTICS
DENOTATIONAL "WHAT"
OPERATIONAL "HOW"
TIME
{NOW}
ALGEBRAIC
(ALG-CBD)
NO LOOPS
WITH LOOPS
ℕ
DISCRETE-TIME
(DT-CBD)
D
ℝ
CONTINUOUS-TIME
(CT-CBD)

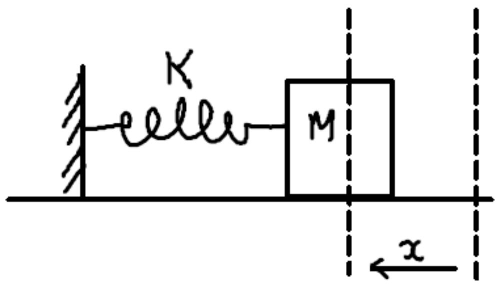


$F = ma$ $\qquad M \frac{d^2x}{dt^2} = -Kx$



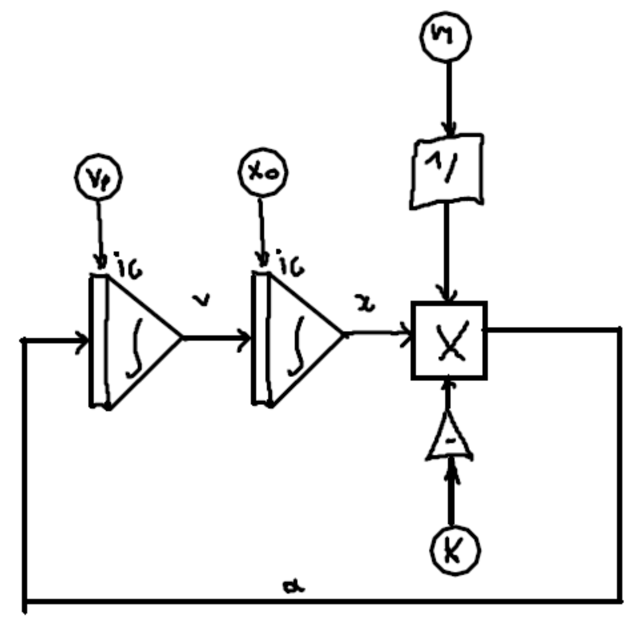
$F = ma$ $\qquad m\dfrac{d^2x}{dt^2} = -Kx$

$$\begin{cases} \dfrac{dx}{dt} = v & x(0) = x_0 \\[2ex] \dfrac{dv}{dt} = -\dfrac{k}{m}x & v(0) = v_0 \end{cases}$$



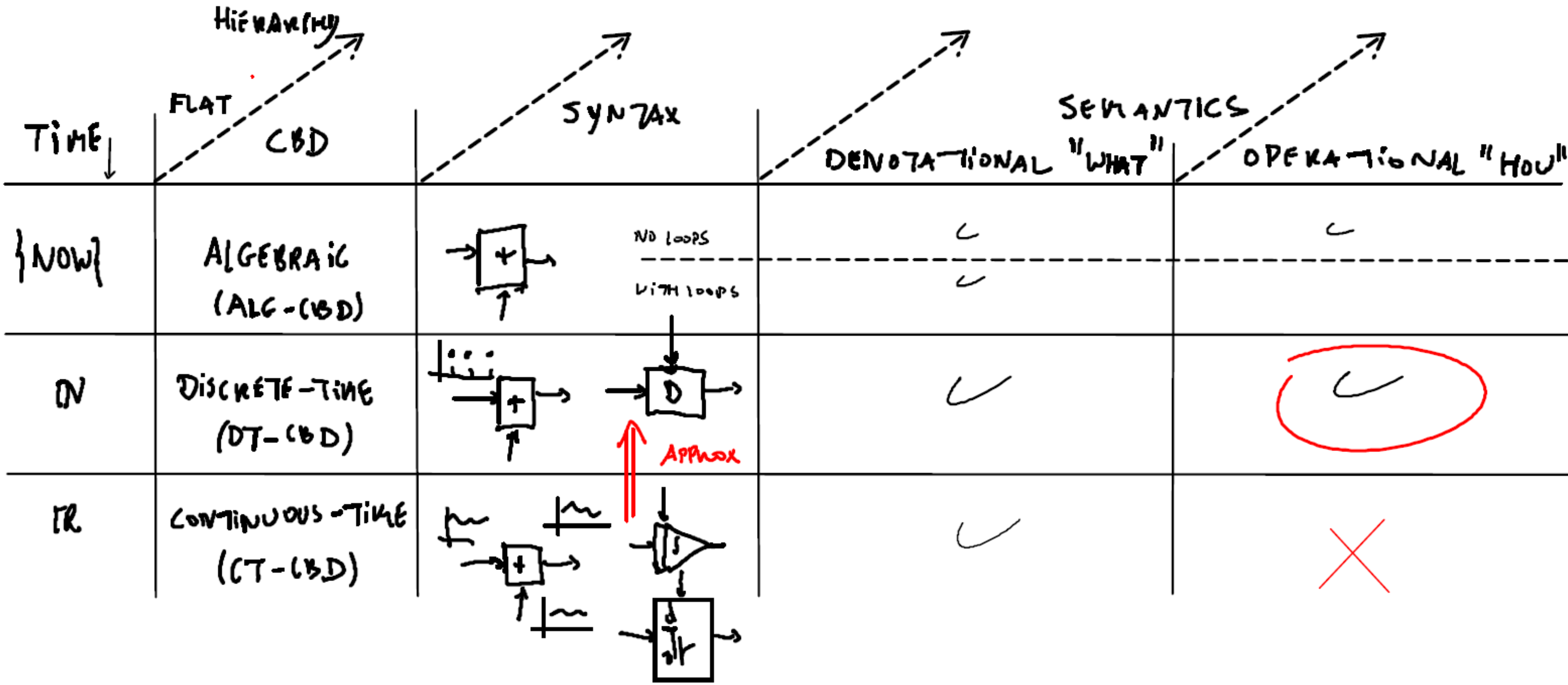
$$\begin{cases} \frac{dx}{dt} = v & x(0) = x_0 \\ \frac{dv}{dt} = -\frac{k}{m}\, x & v(0) = v_0 \end{cases}$$

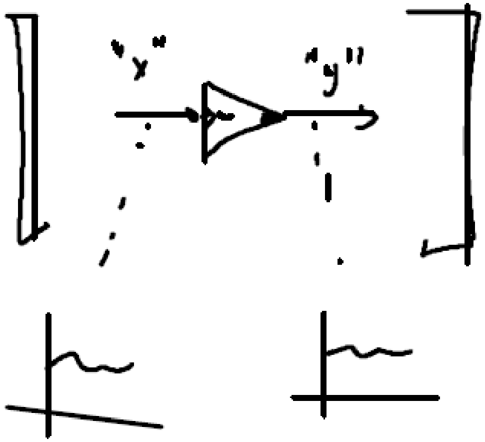
F = ma $\qquad m \frac{d^2x}{dt^2} = -K x$



Physical Systems Modelling

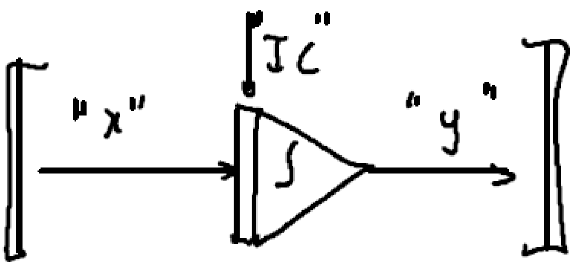
- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure


HIERARCHY
FLAT
CBD
SYNTAX
SEMANTICS
DENOTATIONAL "WHAT"
OPERATIONAL "HOW"
TIME
{NOW}
ALGEBRAIC
(ALG-CBD)
NO LOOPS
WITH LOOPS
ℕ
DISCRETE-TIME
(DT-CBD)
D
APPROX
ℝ
CONTINUOUS-TIME
(CT-CBD)

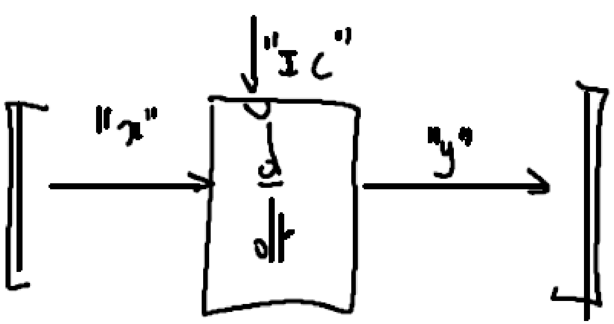


$$:= \quad mvy(t) = -mvx(t), \quad \forall t \in \mathbb{R}$$

$$- : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

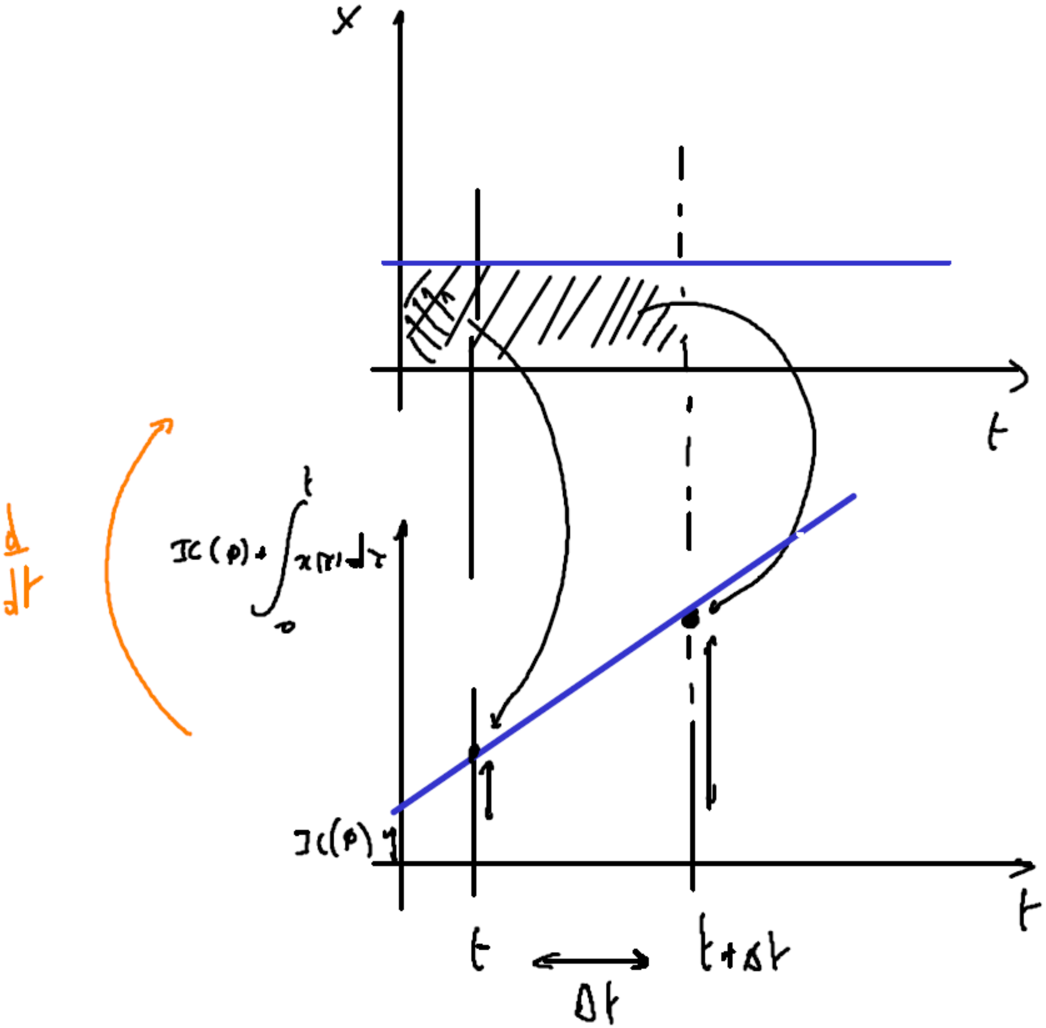


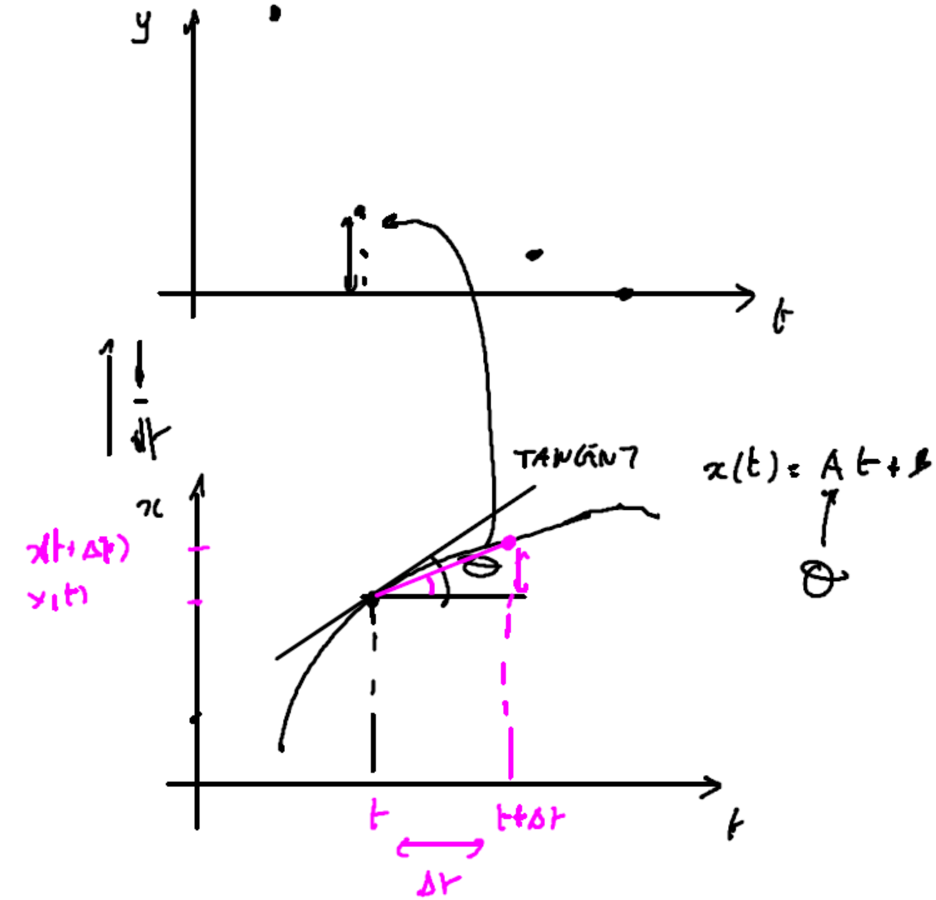
$$:= \quad mvy(t) = \underbrace{mvy(\emptyset)}_{IC(\emptyset)} + \int_0^t x(\tau)\, d\tau, \quad \forall t \in \mathbb{R}$$



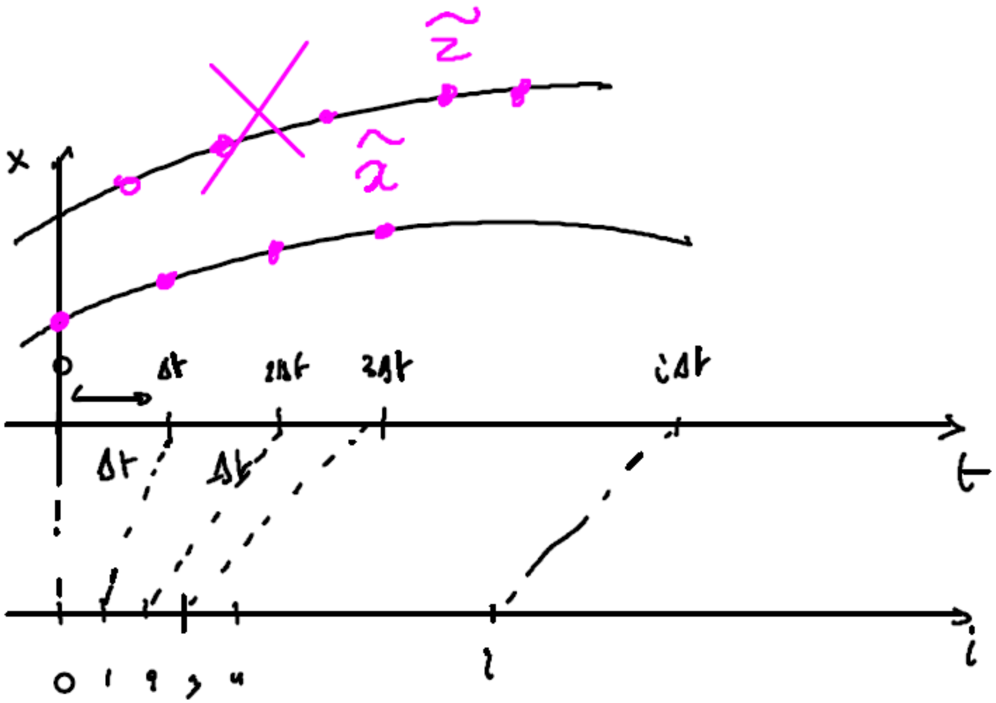
$$:= \quad mvy(t) = \frac{d}{dt} mvx(t), \quad \forall t \in \mathbb{R} \setminus \{\emptyset\}$$

$$mvy(\emptyset) = IC(\emptyset)$$

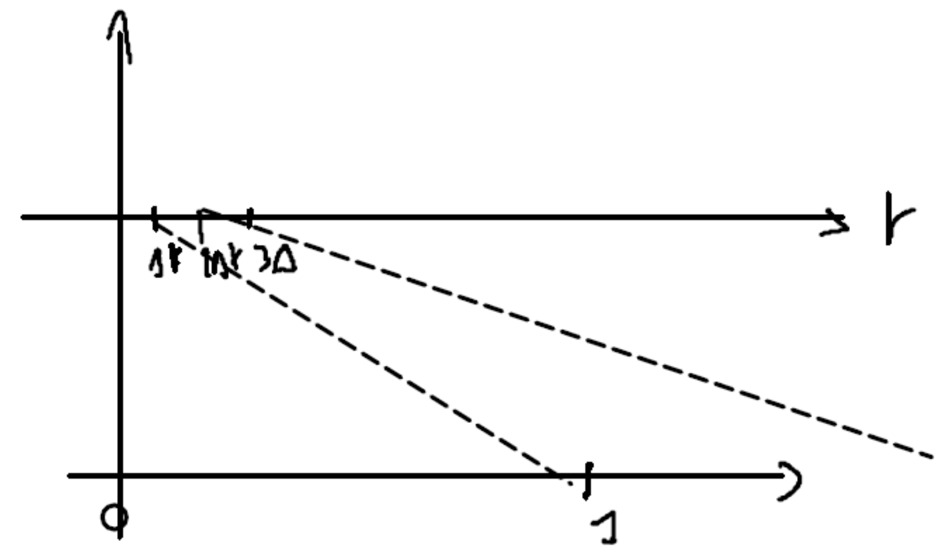




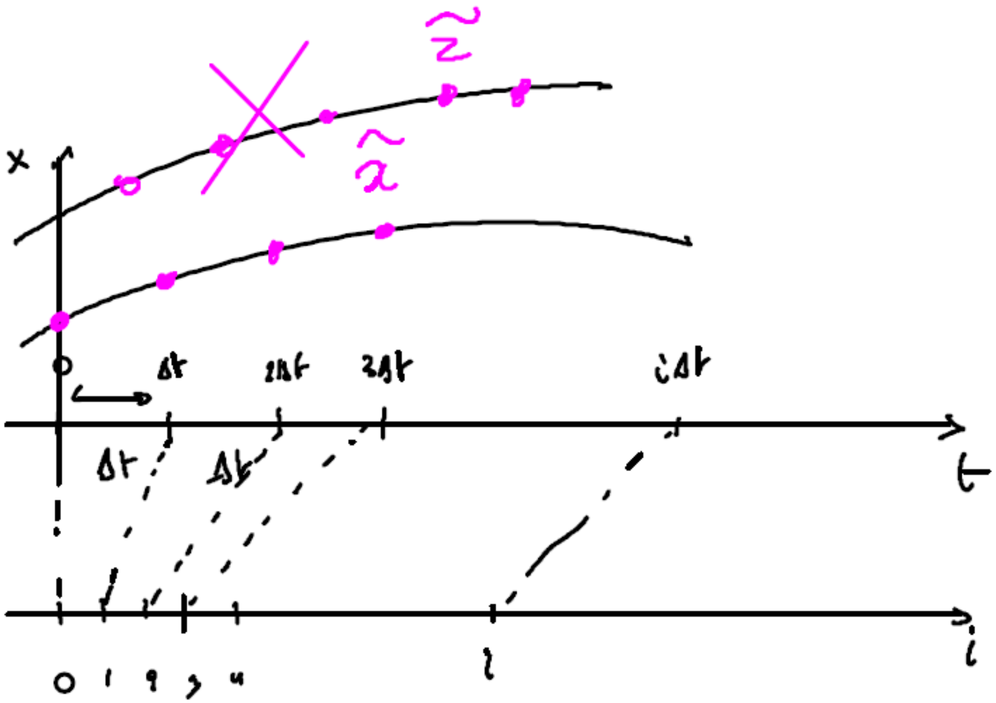
$$\lim_{\Delta t \to 0} \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

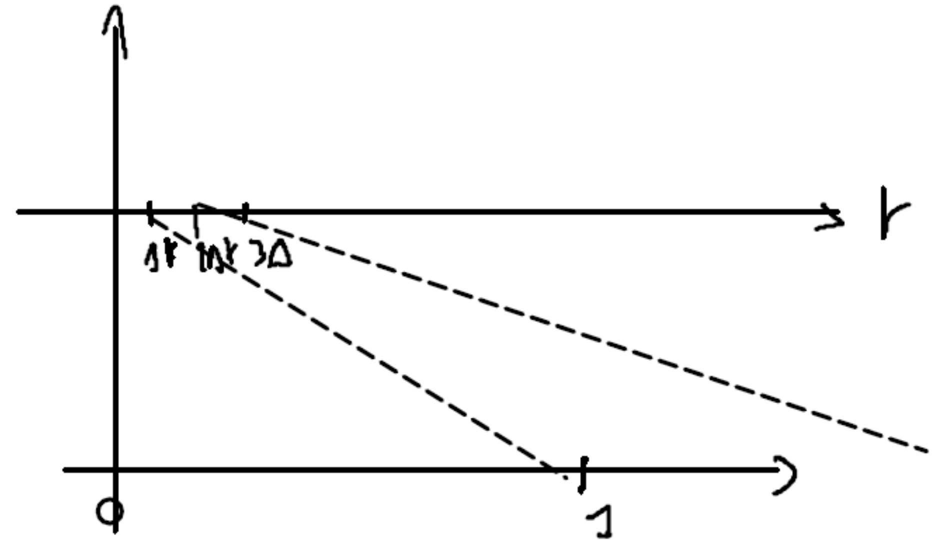


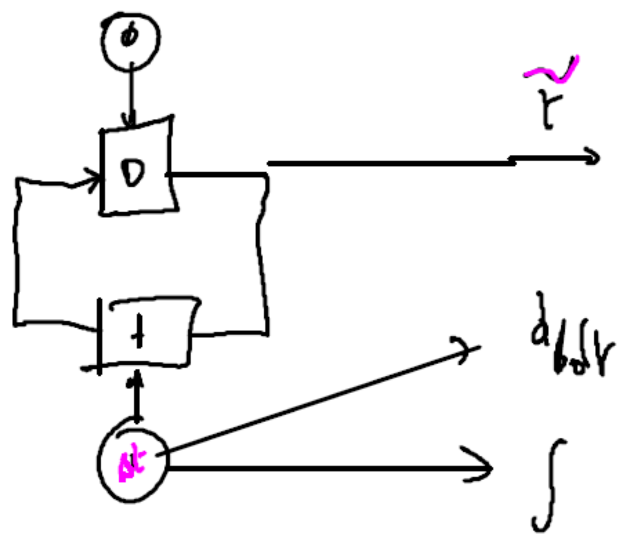
x
$\tilde{z}$
$\tilde{x}$
0
Δt
2Δt
3Δt
iΔt
t
i



r
0
1


x
0
Δt
2Δt
3Δt
iΔt
Δt
Δt
t
0
1
2
3
4
i
i


r
Δt
2Δt
3Δ
0
1


ϕ
D
+
Δt
r̃
∫

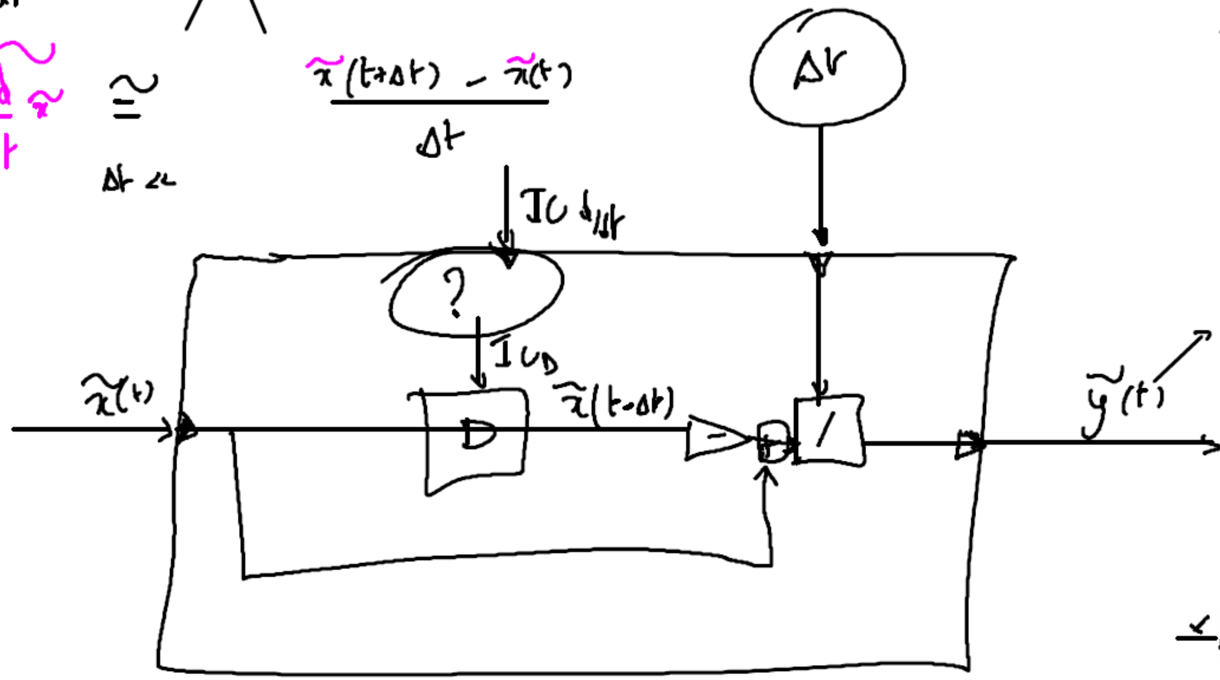
$$\frac{d}{dt}x(t) = \lim_{\Delta t \to 0} \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

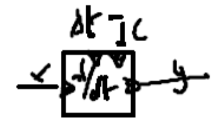
$$\tilde{y}(t) = \widetilde{\frac{d}{dt}x} \simeq \frac{\tilde{x}(t+\Delta t) - \tilde{x}(t)}{\Delta t}$$

$\Delta t \ll$

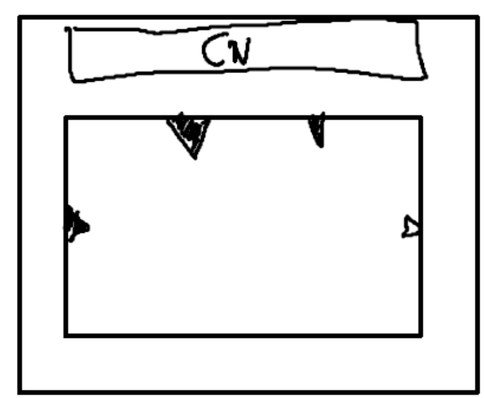
$$\tilde{y}(t+\Delta t) = \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

$$\tilde{y}(t) = \frac{\tilde{x}(t) - \tilde{x}(t-\Delta t)}{\Delta t}$$



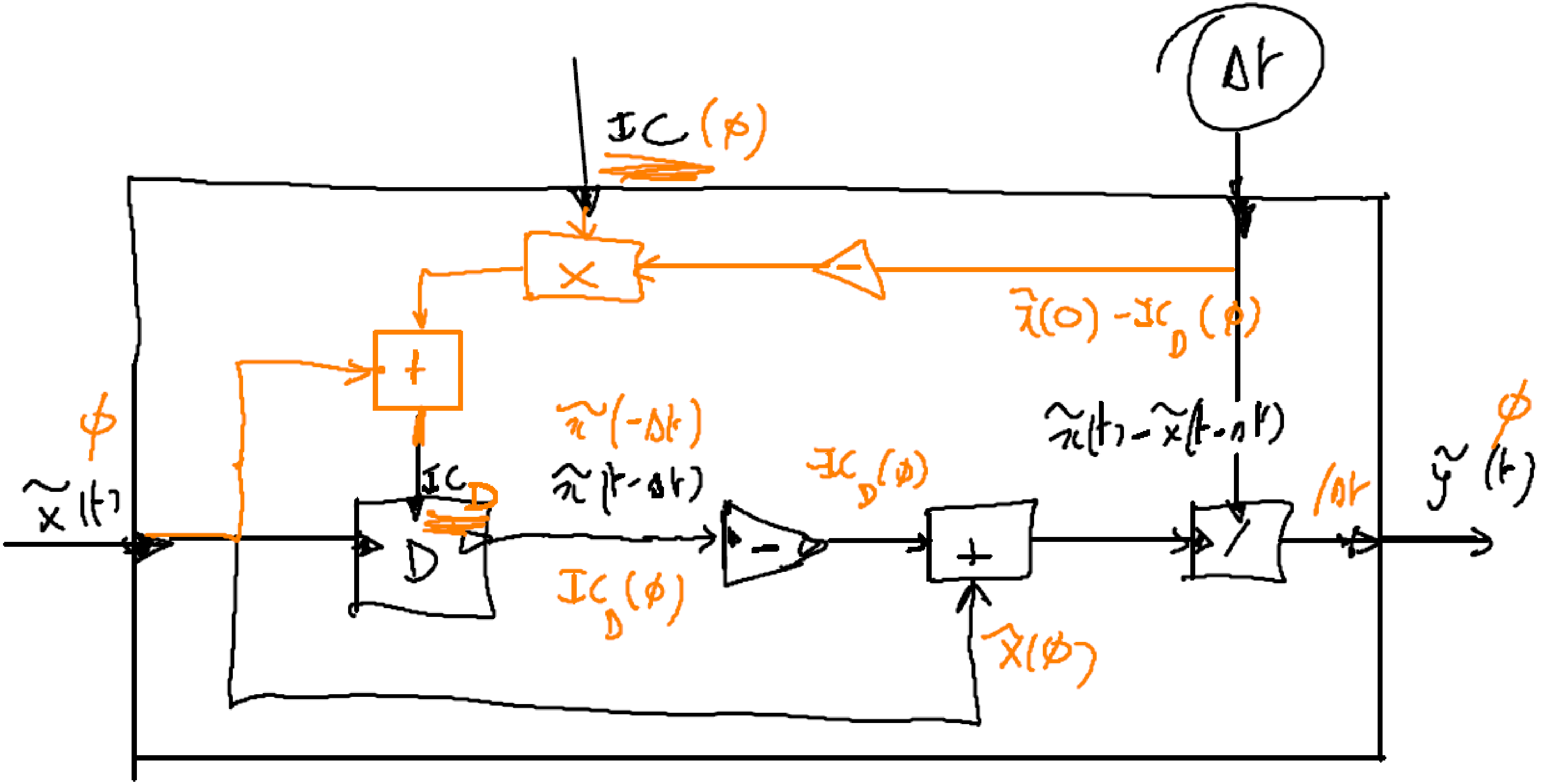


$y = \frac{d}{dt}x \qquad y(0) = IC(0)$



All blocks : = Δt

Synchronous block diagrams



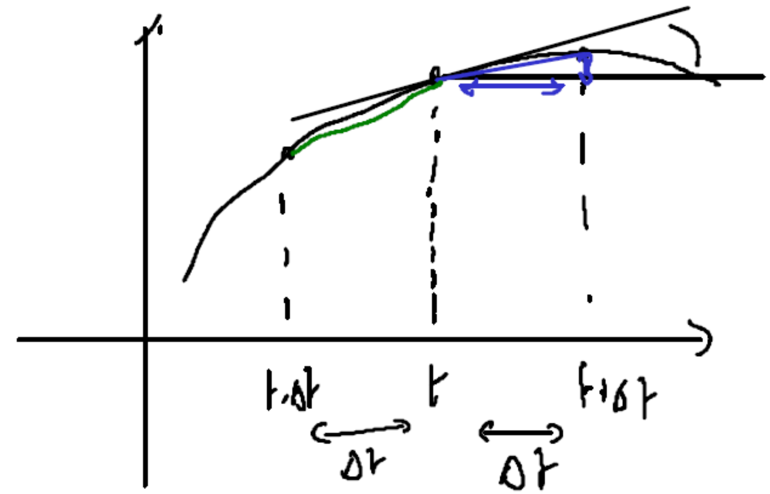
$$\frac{\tilde{x}(0) - IC_D(0)}{\Delta t} = IC_{\frac{d}{dt}}(0) \qquad \tilde{y}(0) = IC_{\frac{d}{dt}}(0)$$

$$\tilde{x}(0) - IC_{\frac{d}{dt}}(0) \cdot \Delta t = IC_D(0)$$

$$\tilde{y}(t) = \frac{\tilde{x}(t) - \tilde{x}(t-\Delta t)}{\Delta t}$$

$$\tilde{y}(t) = \frac{\tilde{x}(t+\Delta t) - \tilde{x}(t)}{\Delta t}$$

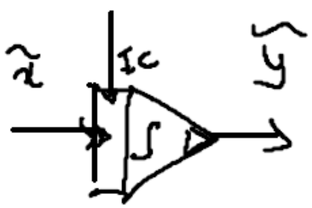
$\Delta t \ll$

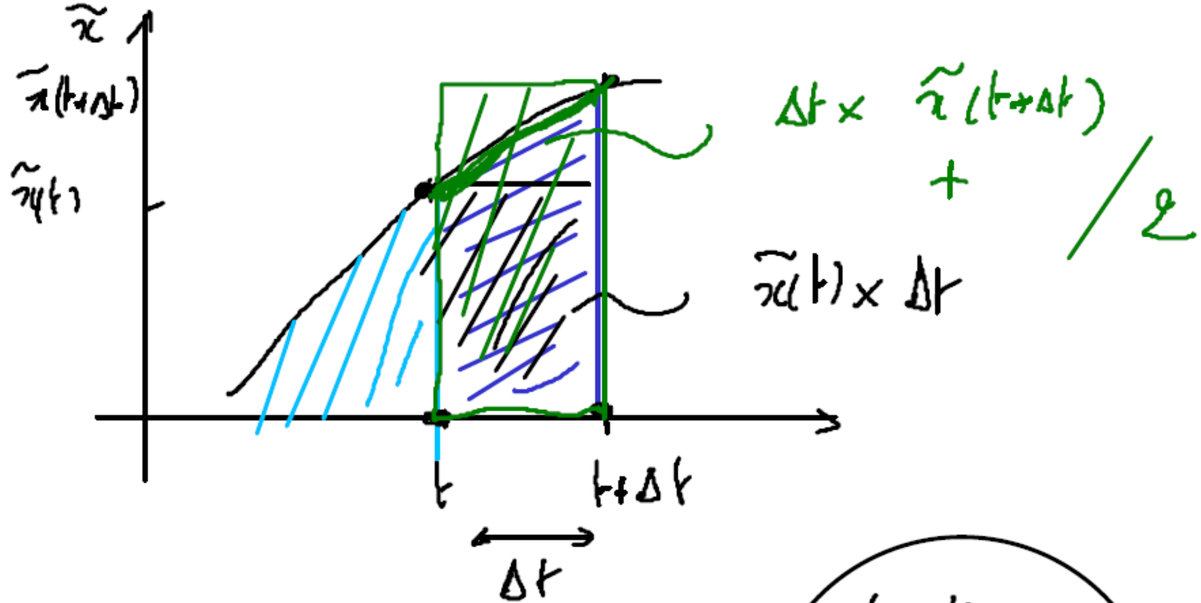


$$\lim_{\Delta t \to 0}$$

$$\tilde{y}(t) = \frac{d\hat{x}}{dt}(t) = \frac{\hat{x}(t+\Delta t) - \hat{x}(t)}{\Delta t}$$

$$\hat{y}(t) = \frac{d\tilde{x}}{dt}(t) = \frac{\tilde{x}(t) - \tilde{x}(t-\Delta t)}{\Delta t}$$



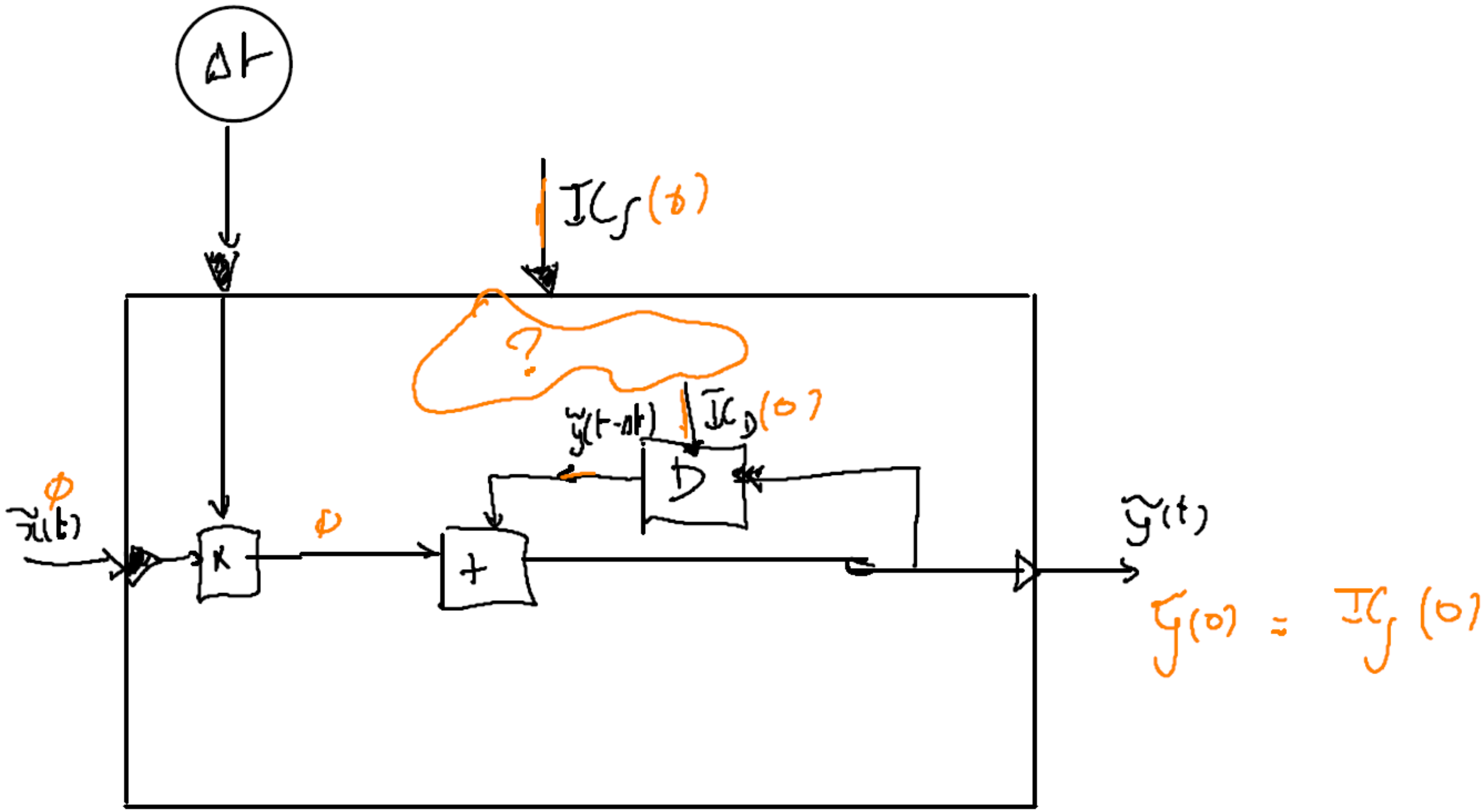


$$\tilde{y}(t+\Delta t) = \tilde{y}(t) + \int_{t}^{t+\Delta t} \tilde{x}(\tau)\,d\tau$$

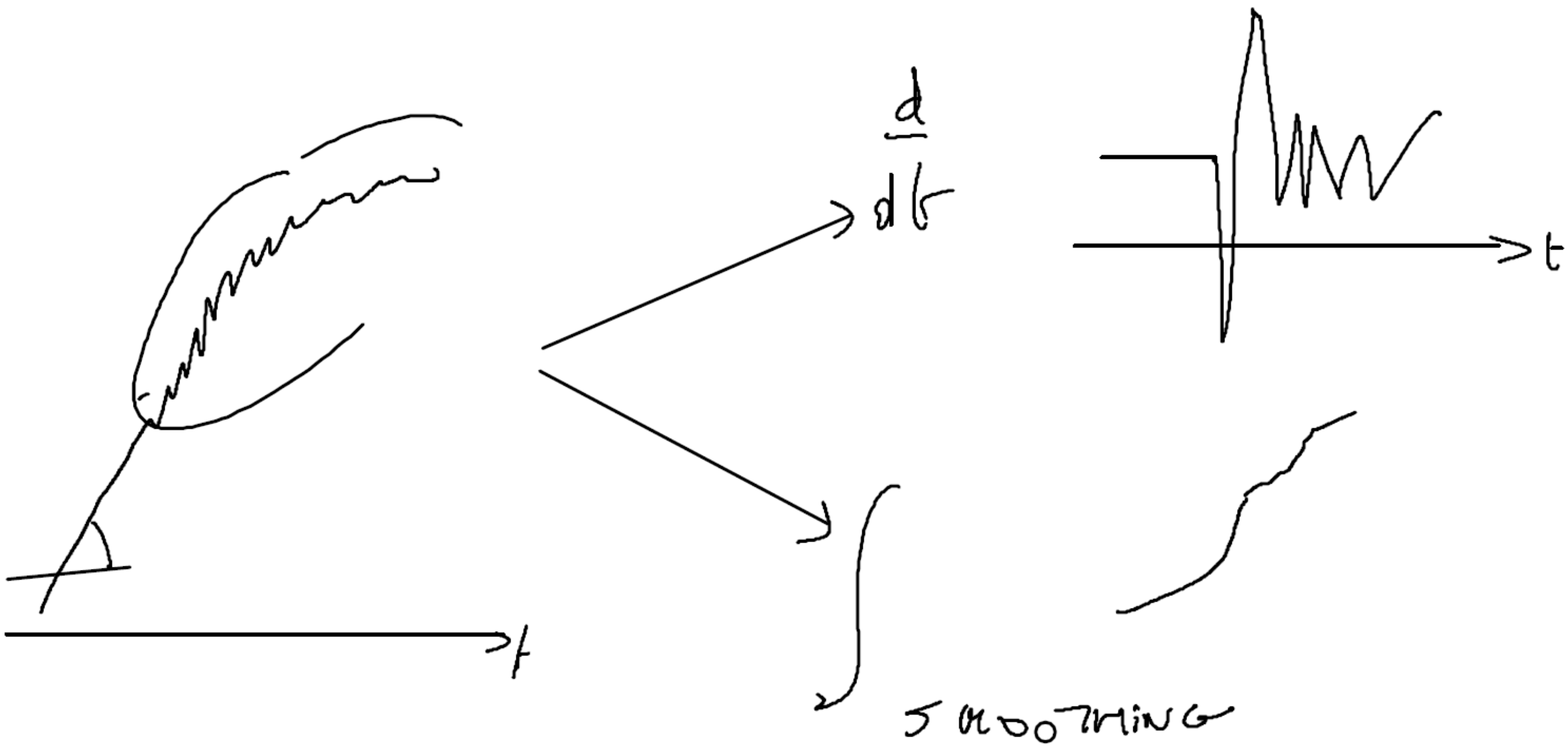
$$\lim_{\Delta t \to 0} \qquad \Delta t\,\tilde{x}(t) \qquad \Delta t\,\tilde{x}(t+\Delta t)$$

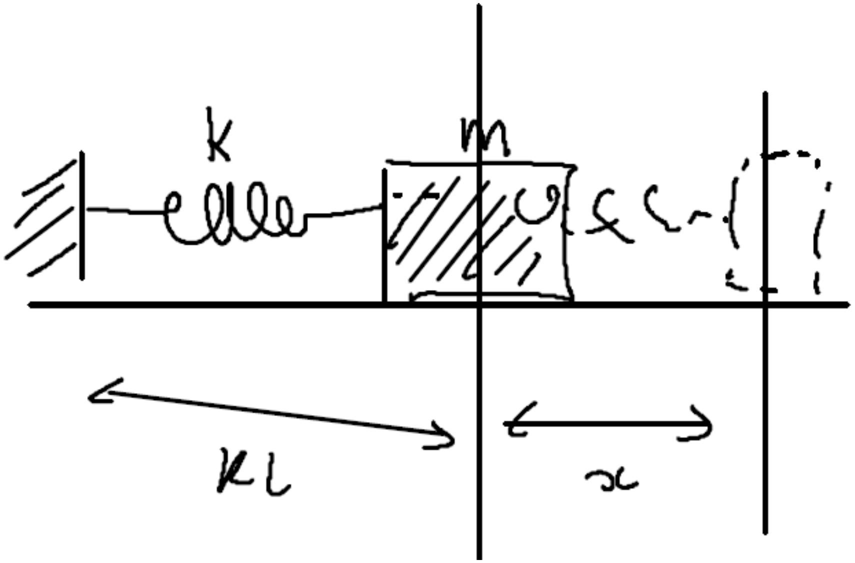
$$\tilde{y}(t) = y(t-\Delta t) + \Delta t\,\tilde{x}(t)$$

APPROX



Δt
IC_f (0)
?
ỹ(t-Δt)
IC_D (0)
D
φ
x̃(t)
φ
K
+
ỹ(t)
ỹ(0) = IC_f (0)

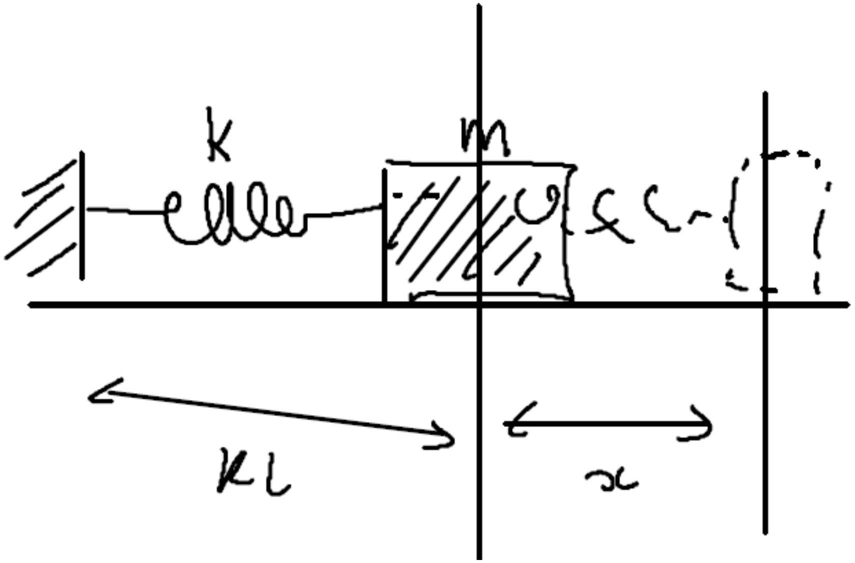

t
d/dt
t
SMOOTHING



HOOKE'S LAW

$\underline{F} = -kx$

$x \ll$



HOOKE'S LAW

$\underline{F} = -kx$ $\quad x \ll$

$F = ma$

$\underline{F} = m \frac{dv}{dt} = m \frac{d^2 x}{dt^2}$

$v = \frac{dx}{dt}$

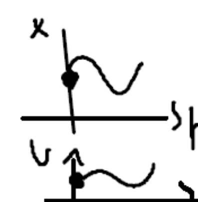
$\frac{dv}{dt} = -\frac{k}{m} x, \; v(0)$

$\frac{dx}{dt} = v, \; x(0)$

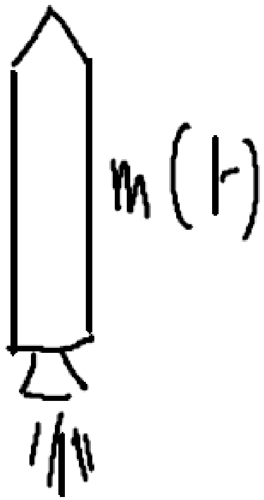
ODE
ORDINARY DIFFERENTIAL EQNS

PDE

$x(t), \; v(t)$



F = m **a** is <u>not</u> "rocket science"

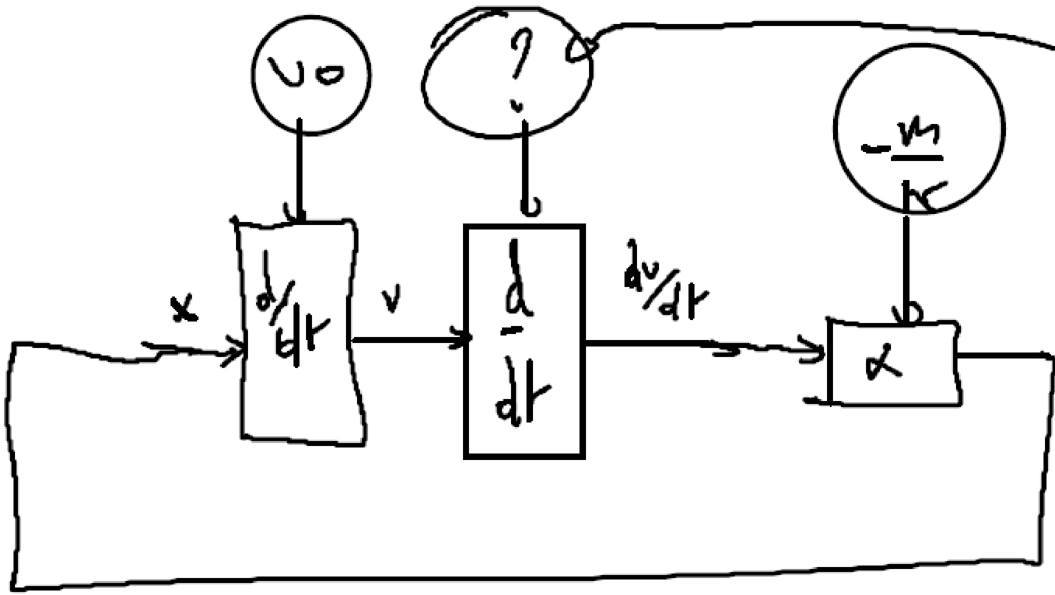


$$v = \frac{d}{dt}x$$

$$a = \frac{d}{dt}v = \frac{d^2}{dt^2}x$$

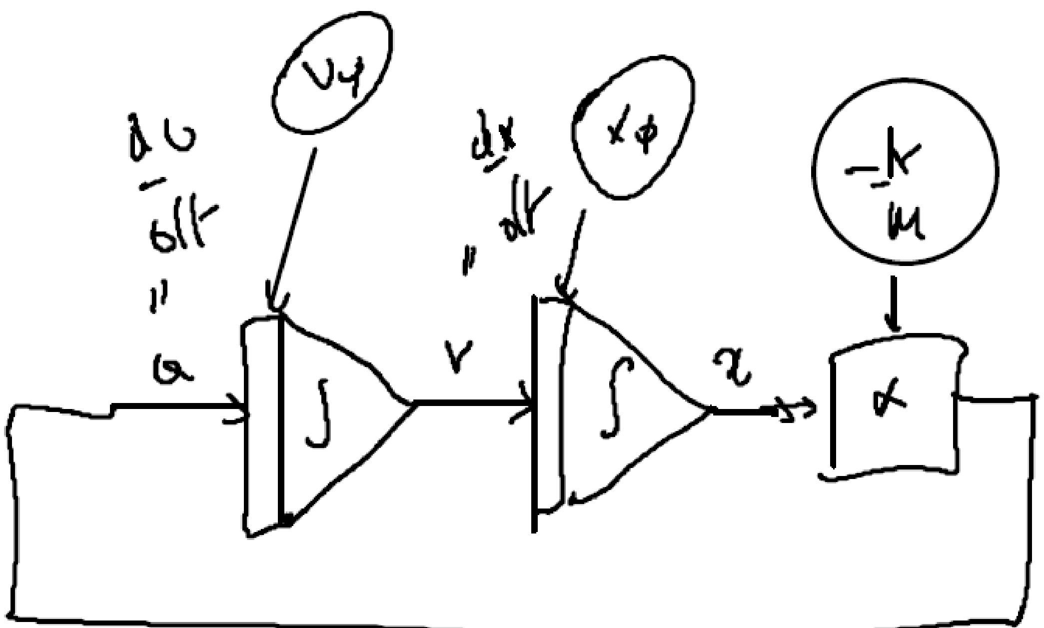
$$F = \frac{dp}{dt} = \frac{d}{dt}(mv)$$

$$= \underbrace{\frac{d}{dt}m \cdot v} + m\underbrace{\frac{dv}{dt}}_{a}$$

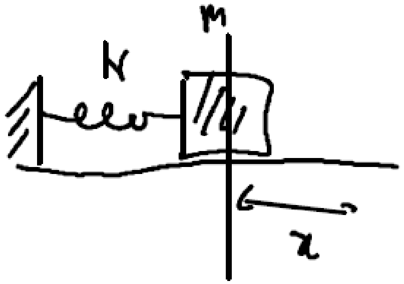


$$x = -\frac{m}{k}\frac{dv}{dt}$$

$$\frac{dv}{dt}(0) = -\frac{k}{m}x(0)$$

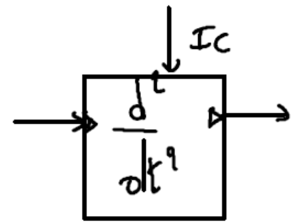


$$\lim_{\Delta t \to \phi}$$

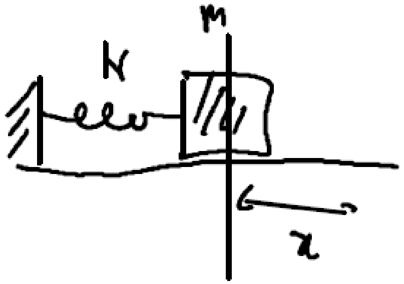


HARMONIC EQN.

$$\frac{d^2x}{dt^2} = -x, \quad x(0) = x_0, \quad \frac{dx}{dt}(0) = v(0) = v_0 \qquad \text{ODE}$$

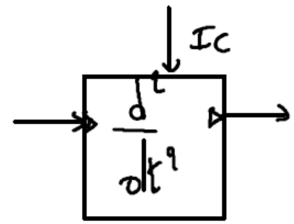


$$\begin{cases} \dfrac{dx}{dt} = v, & x(0) = x_0 \\ \dfrac{dv}{dt} = -x, & v(0) = v_0 \end{cases}$$



HARMONIC EQN.

$$\frac{d^2x}{dt^2} = -x, \quad x(0) = x_0, \quad \frac{dx}{dt}(0) = v(0) = v_0 \qquad \text{ODE}$$



$$\begin{cases} \dfrac{dx}{dt} = v, & x(0) = x_0 \\[2ex] \dfrac{dv}{dt} = -x, & v(0) = v_0 \end{cases}$$

HIGHER-ORDER ODES

$$\frac{d^k x}{dt^k} = \overbrace{\frac{d}{dt}\left(\frac{d}{dt}\left(\cdots \frac{d}{dt} x\right)\right)}^{k}$$

$$\begin{cases} \dfrac{dx}{dt} = x\_der\_1 \\[2ex] \dfrac{d\, x\_der\_1}{dt} = x\_der\_2 \\[2ex] \vdots \\[1ex] \dfrac{d\, x\_der\_{k-1}}{dt} = x\_der\_k \end{cases}$$

$$\frac{d^2x}{dt^2} = -x \quad , \quad x(0) = \phi \quad , \quad \frac{dx}{dt}(0) = 1$$

$$\frac{dx}{dt} = x$$

$$x(t) = e^x + c$$

$$\begin{cases} x(t) = A \sin(t) + B \cos(t) \left(+ c \right) \\ x(0) = \phi \quad , \quad \frac{dx}{dt}(0) = 1 \end{cases}$$

$$\frac{dx}{dt} = A \cos(t) - B \sin(t)$$

$$\frac{d^2x}{dt^2} = -A \sin(t) - B \cos(t) = -\left(A \sin(t) + B \cos(t)\right) = -x(t) \qquad \text{q.e.d.}$$

$$x(0) = \quad B = \phi$$

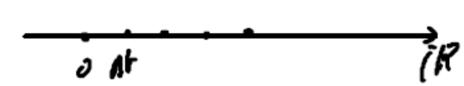
$$\frac{dx}{dt}(0) = \quad A = 1$$

$$\begin{cases} x(t) = \sin(t) \\ \frac{dx}{dt}(t) = \cos(t) \end{cases}$$

$$\tilde{x}(\hat{t})$$

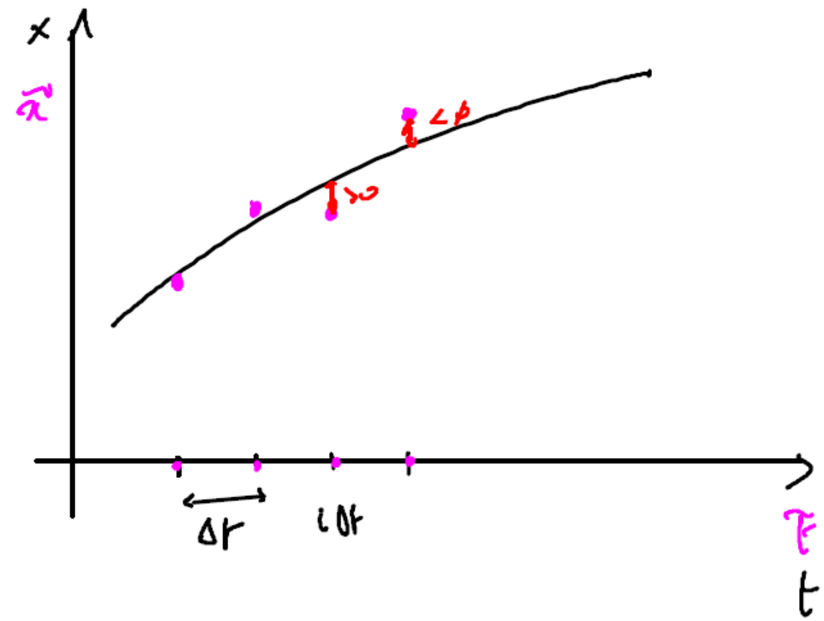
$$\widetilde{\frac{dx}{dt}} = \tilde{v}(\hat{t})$$

$$\hat{t} = i \times \Delta t$$



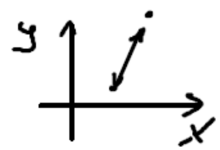
ERROR ~ $e(i\Delta t) = x(i\Delta t) - \tilde{x}(i\Delta t)$

TOTAL ERROR ~ $\sum_{i=0}^{i\_END} e(i\Delta t)$



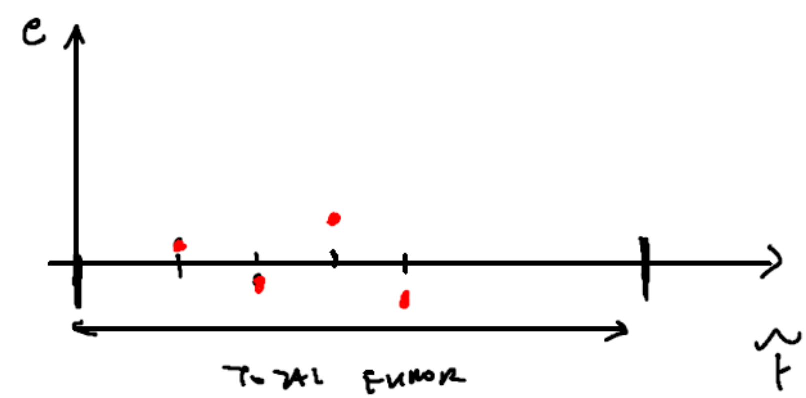
$$e(i\Delta t) = \left| x(i\Delta t) - \tilde{x}(i\Delta t) \right|$$

$$e(i\Delta t) = \left(x(i\Delta t) - \tilde{x}(i\Delta t) \right)^2$$



"SUM OF SQUARES"

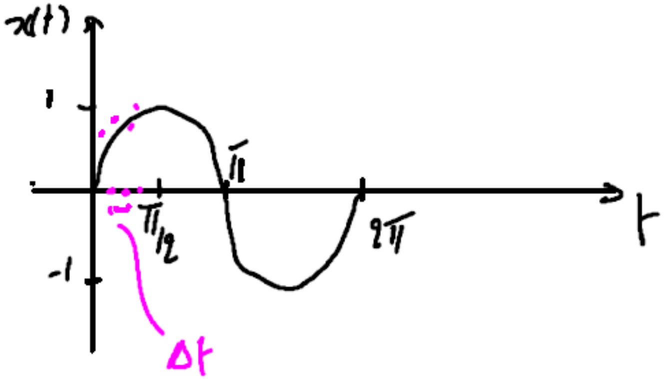
$$\text{TOTAL ERROR} = \sum_{i=\phi}^{i\_END} \left(x(i\Delta t) - \tilde{x}(i\Delta t) \right)^2$$

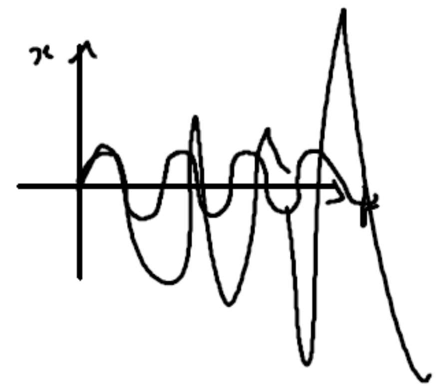


$$= \int \left(x(\tau) - \tilde{x}(\tau) \right)^2 d\tau$$

sin(τ) CBD

CBD



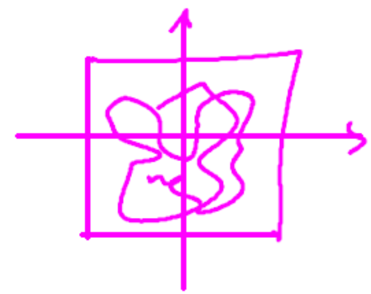


UNSTABLE

SYSTEM ?

NUMERICAL ?

BOUNDED REGION



PHASE PLOT

~ STABILITY

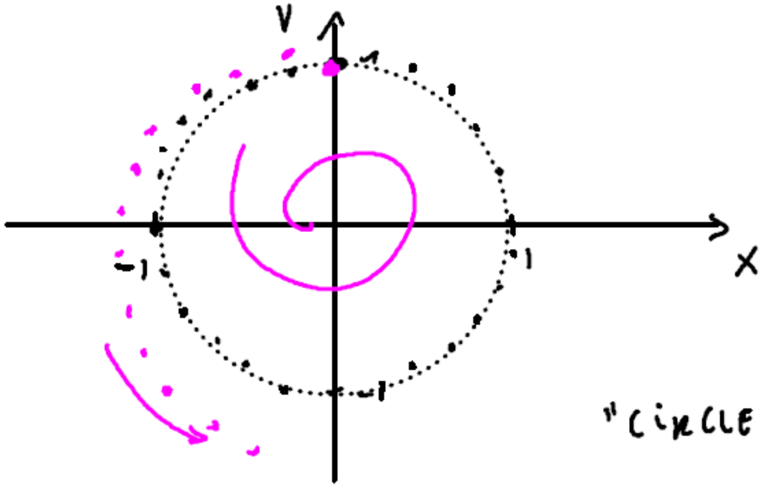
- SYSTEM (SYS)
- NUMERICAL

~ Δt, SCHEME

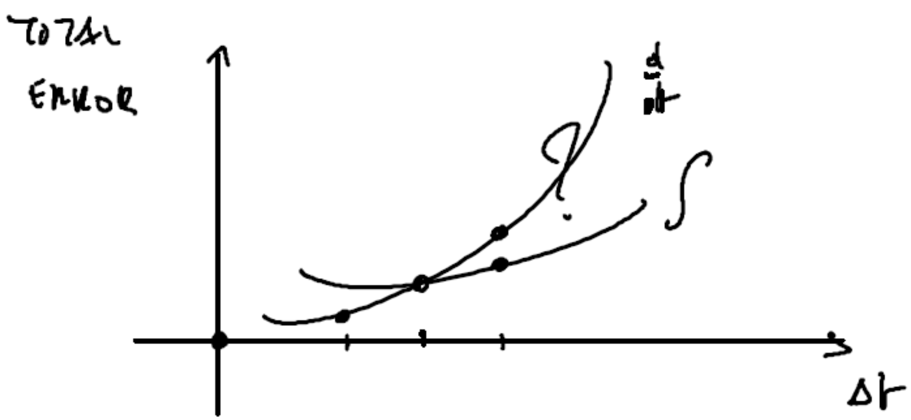
$(x(t), v(t))$

$x(0) = 0 \quad , \quad v(0) = 1$

$x(\frac{\pi}{2}) = \quad , \quad v(\frac{\pi}{2}) =$



"CIRCLE TEST"



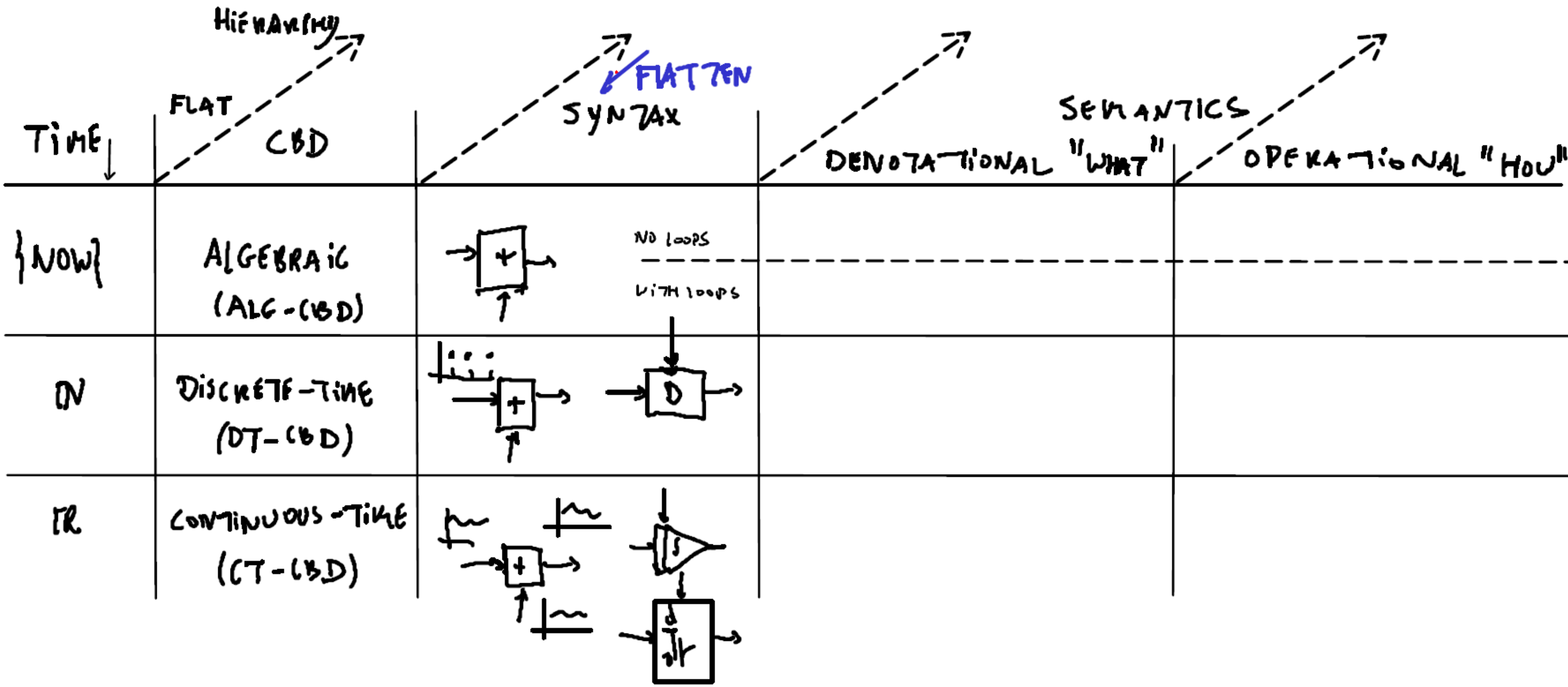
DISCRETIZATION
SCHEME
FIXED

TOTAL ERROR (Δt) = $\overset{?}{f}(\Delta t)$

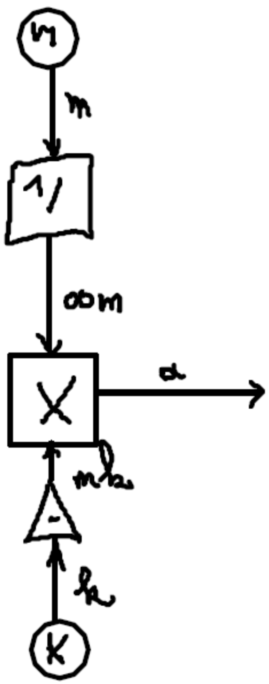
$$x(t+\Delta t) = x(t) + \frac{x'(t)\,\Delta t}{1!} \Big| + \frac{x''(t)}{2!}\Delta t^2 + \frac{x^{(3)}(t)}{3!}\Delta t^3 + \dots$$

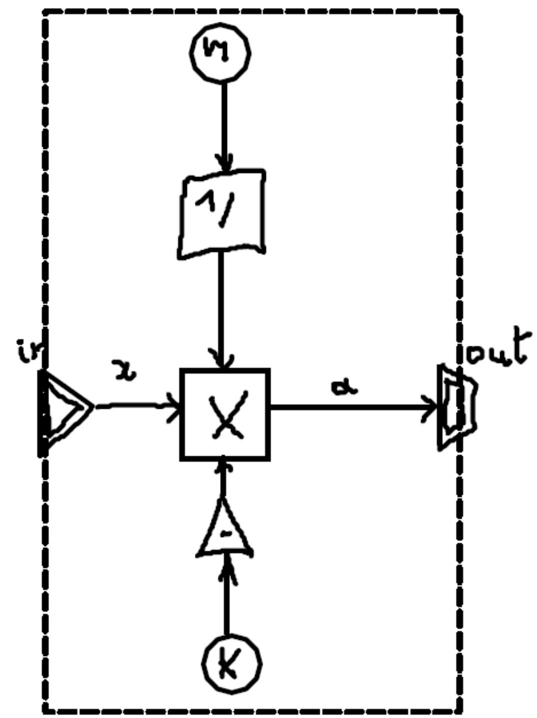
TAYLOR EXPANSION

$\longrightarrow \mathcal{O}(\Delta t^?)$

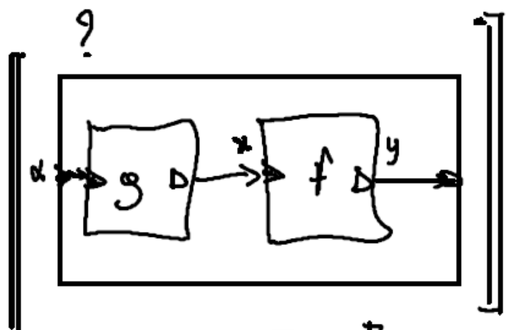


HIERARCHY
FLAT
CBD
TIME
SYNTAX
FLATTEN
SEMANTICS
DENOTATIONAL "WHAT"
OPERATIONAL "HOW"
{NOW}
ALGEBRAIC
(ALG-CBD)
NO LOOPS
WITH LOOPS
ℕ
DISCRETE-TIME
(DT-CBD)
D
ℝ
CONTINUOUS-TIME
(CT-CBD)

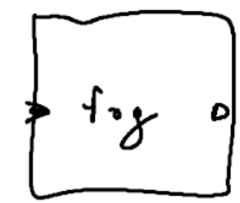

m
m
1/
X
K


m
1/
in
X
out
K

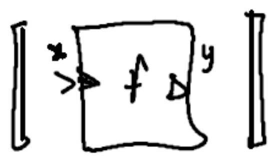
HIERARCHY



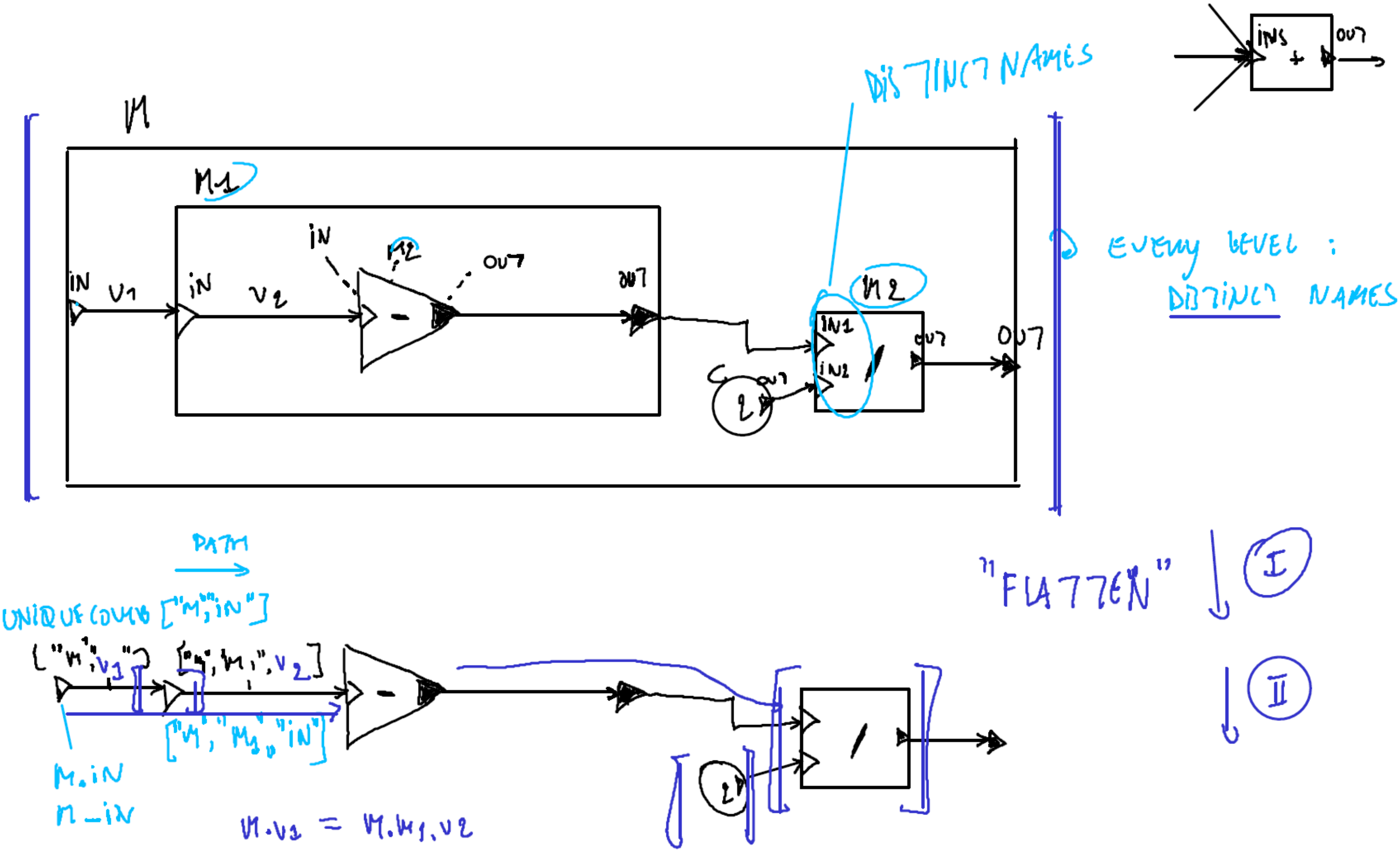
$$= f(g(\alpha)) = (f \circ g)(\alpha)$$



FUNCTION BLOCK



$$= \quad y := f(x)$$

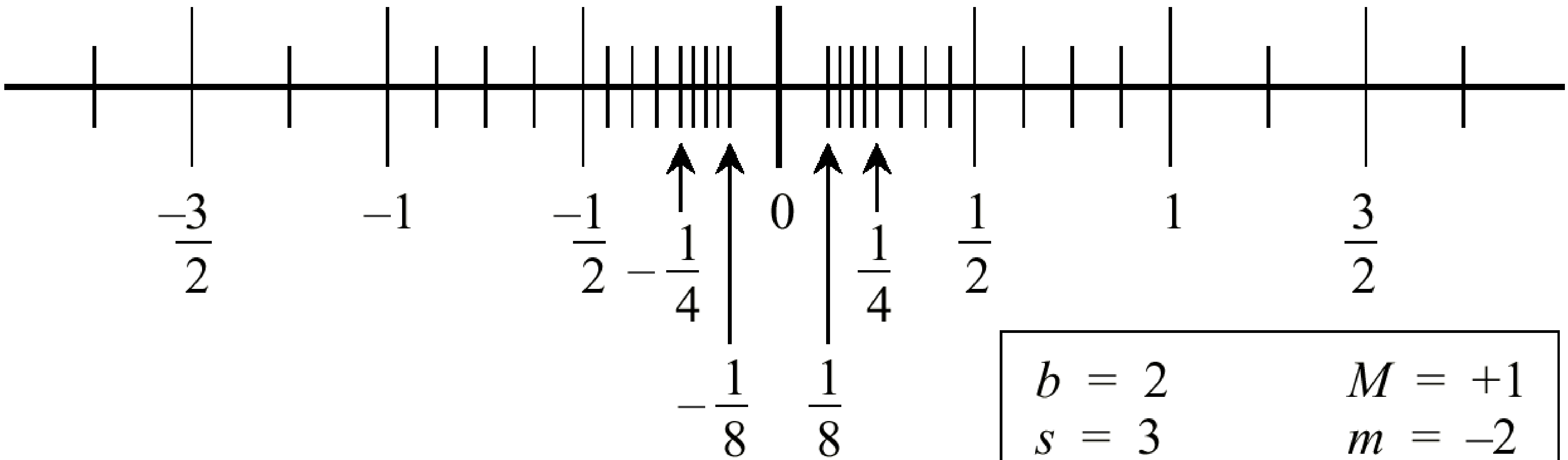


DISTINCT NAMES
INS
OUT
M
M1
M2
IN
V1
V2
OUT
IN1
IN2
2
EVERY LEVEL : DISTINCT NAMES
"FLATTEN"
I
II
PATH
UNIQUE COMB ["M","IN"]
["M","V1"]
["M","M1","V2"]
["M","M1","IN"]
M.IN
M_IN
M.V1 = M.M1.V2

Physical Systems Modelling

- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

Example Floating Point Format



$b = 2 \qquad M = +1$

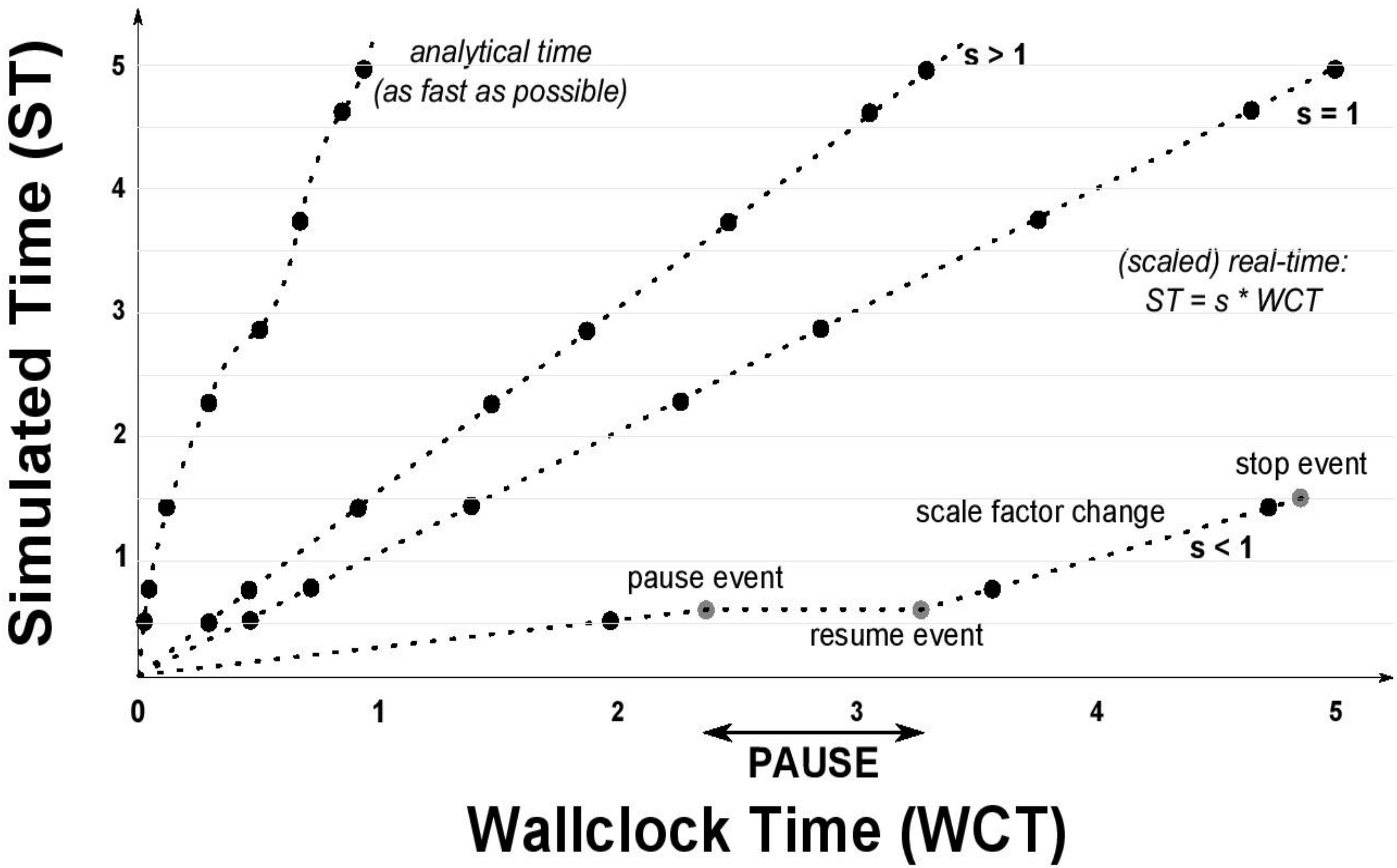
$s = 3 \qquad m = -2$

- Smallest non-zero positive number = $\boldsymbol{b^m \times b^{-1}}$ = 1/8
- Largest non-zero positive number = $\boldsymbol{b^M \times (1 - b^{-s})}$ = 7/4
- Smallest gap = $\boldsymbol{b^m \times b^{-s}}$ = 1/32
- Largest gap = $\boldsymbol{b^M \times b^{-s}}$ = 1/4
- Number of representable numbers = $\mathbf{2x((M\text{-}m)+1)x(b\text{-}1)xb^{s-1}+1}$ = 33

  ... fits into available bits? Optimal number of bits?
- Note: fill the gap around 0: **de-normalized**

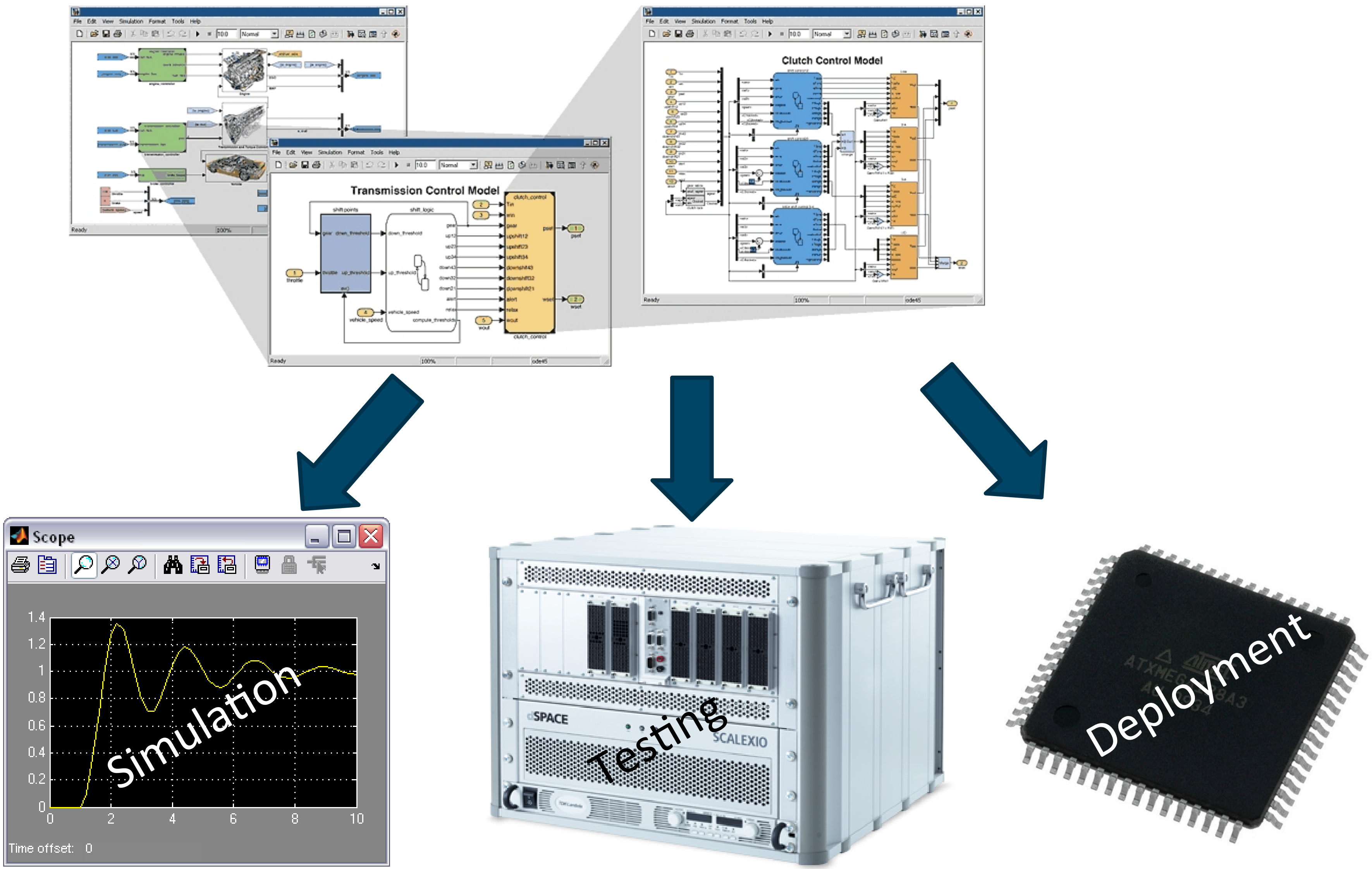
Physical Systems Modelling

- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

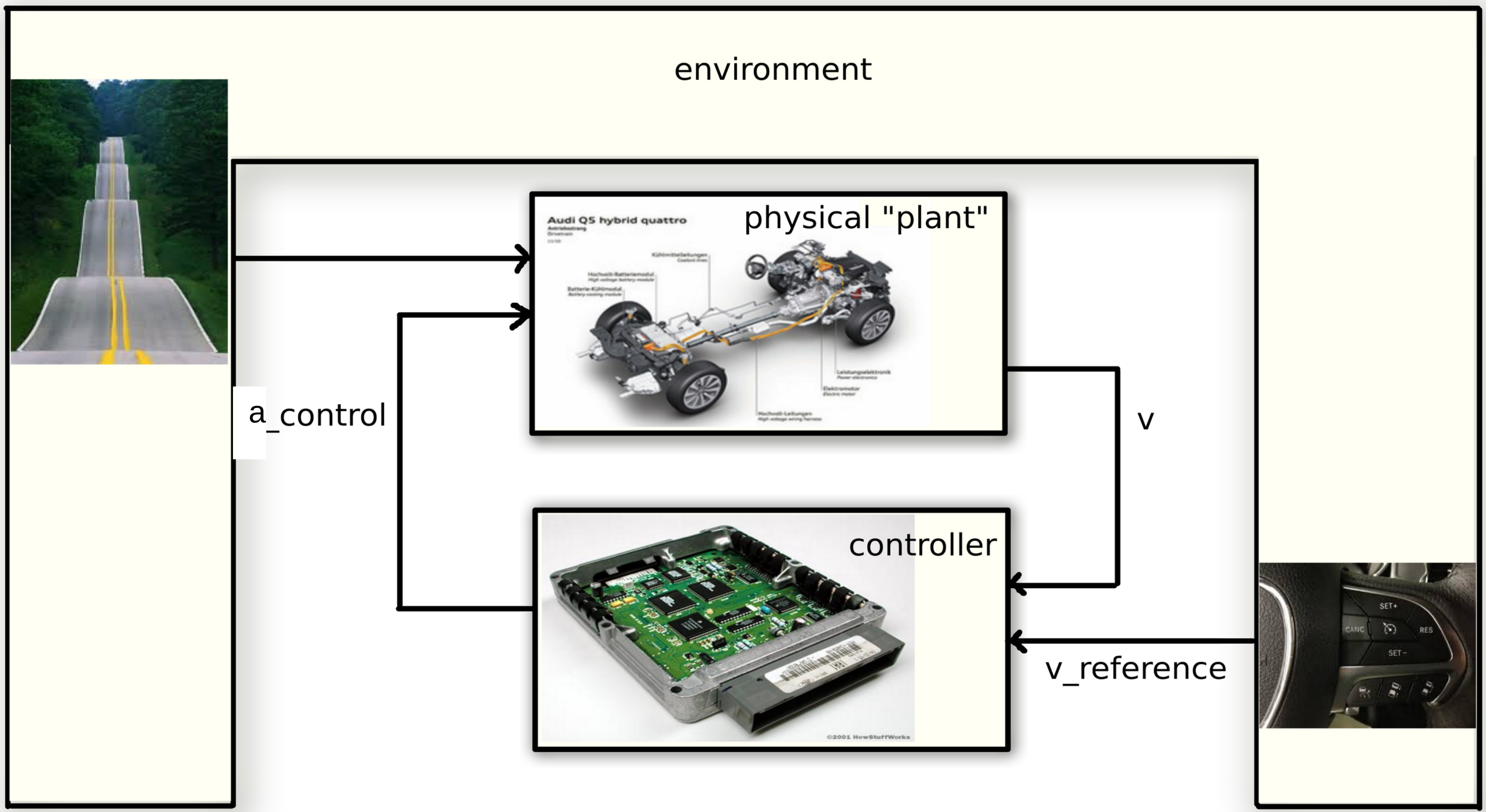


analytical time
(as fast as possible)
s > 1
s = 1
(scaled) real-time:
ST = s * WCT
stop event
scale factor change
s < 1
pause event
resume event
PAUSE
Simulated Time (ST)
Wallclock Time (WCT)
0
1
2
3
4
5

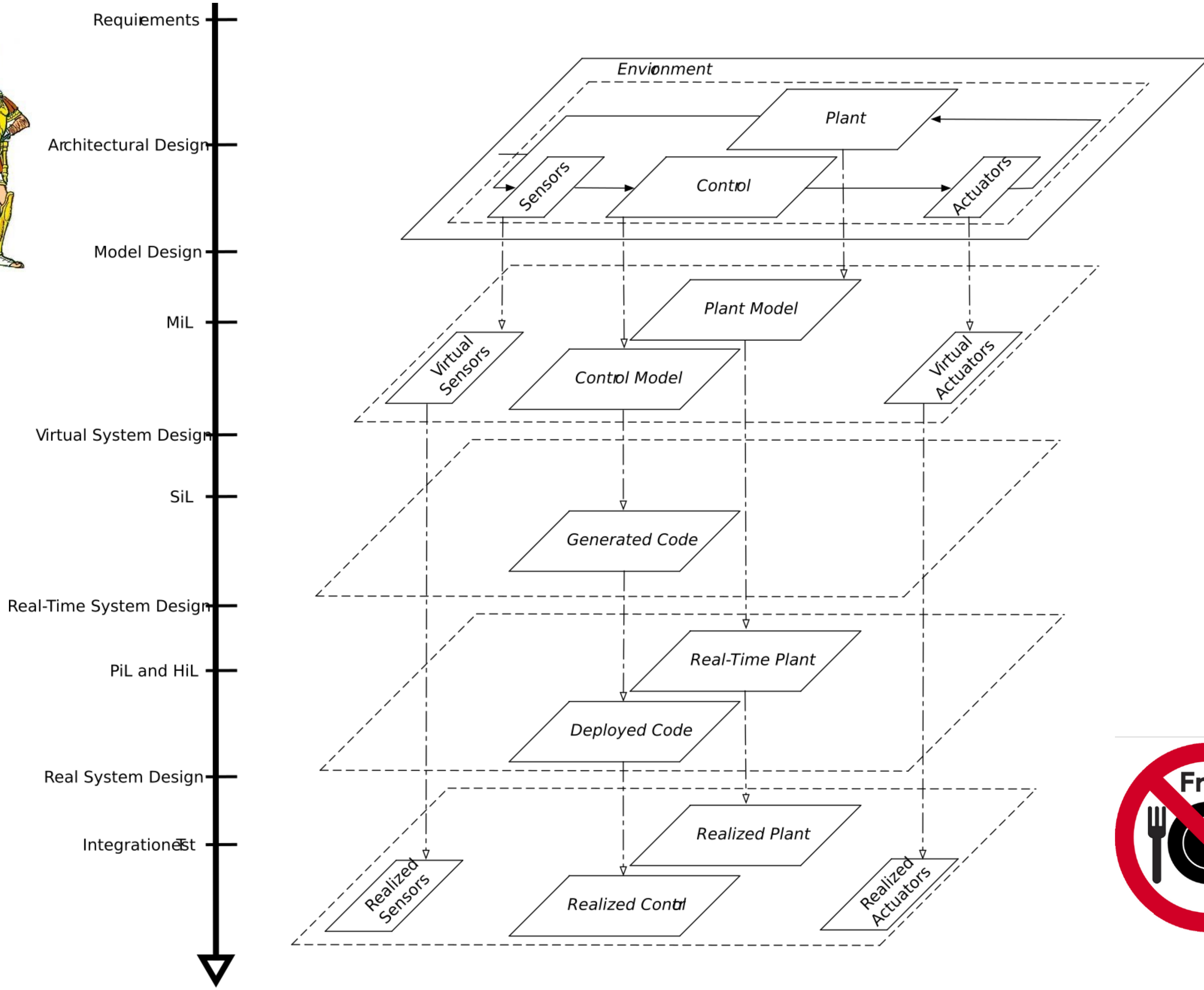
Model-Based System Design



MiL, HiL, SiL, ...


environment
physical "plant"
Audi Q5 hybrid quattro
a_control
v
controller
v_reference

XiL: X = Model, Software, Processor, Hardware



vertical consistency!

Ken Vanherpen. A contract-based approach for multi-viewpoint consistency in the concurrent design of cyber-physical systems. PhD thesis University of Antwerp. 2018.

Physical Systems Modelling

- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

(*)

$y = c_1 x + c_2 x^2$

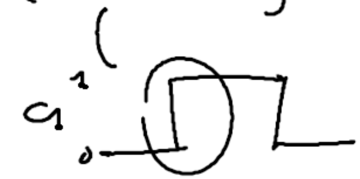
	c_1	c_2
MODE 1	1	0
MODE 2	0	1

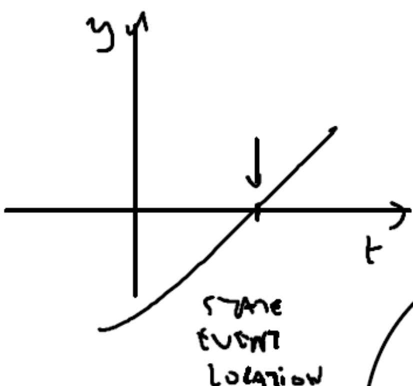
$c_1 + c_2 = 1 \qquad c_1, c_2 \in \{0, 1\}$

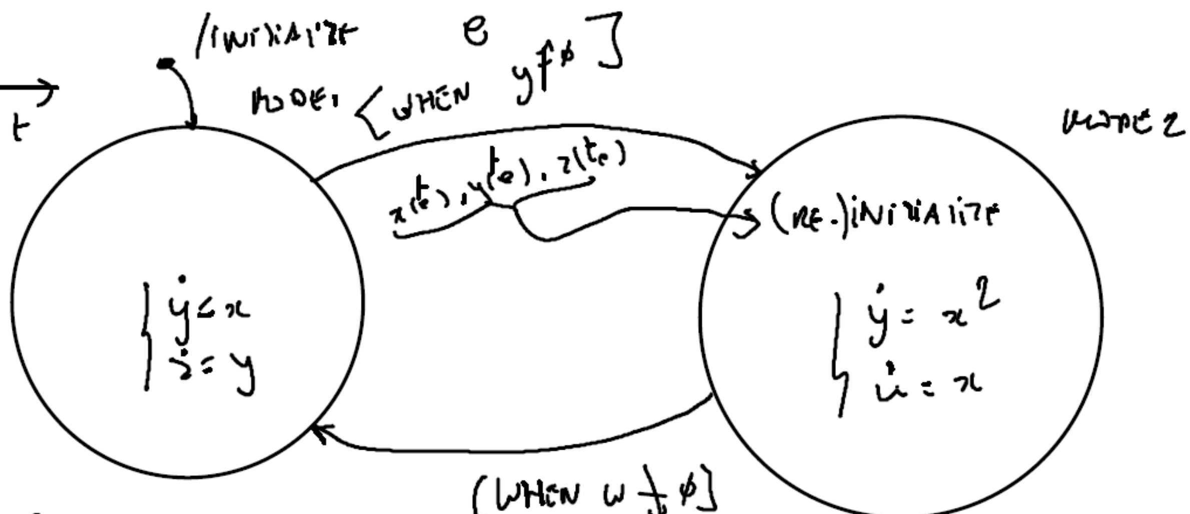
- MISS OPPORTUNITIES (e.g. LIN. ALG. LOOPS)
- NUMERICAL PROBLEMS (CONTINUITY)

$y = x$

$y = x^2$



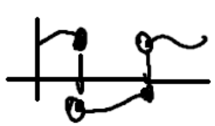


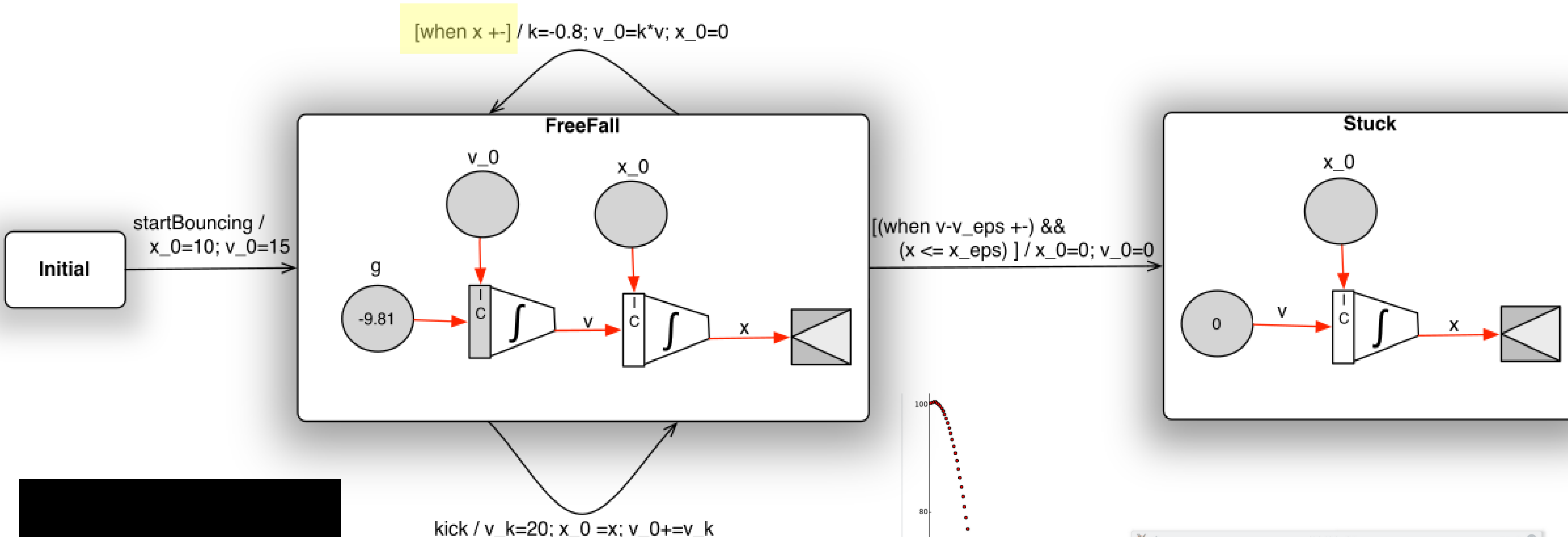


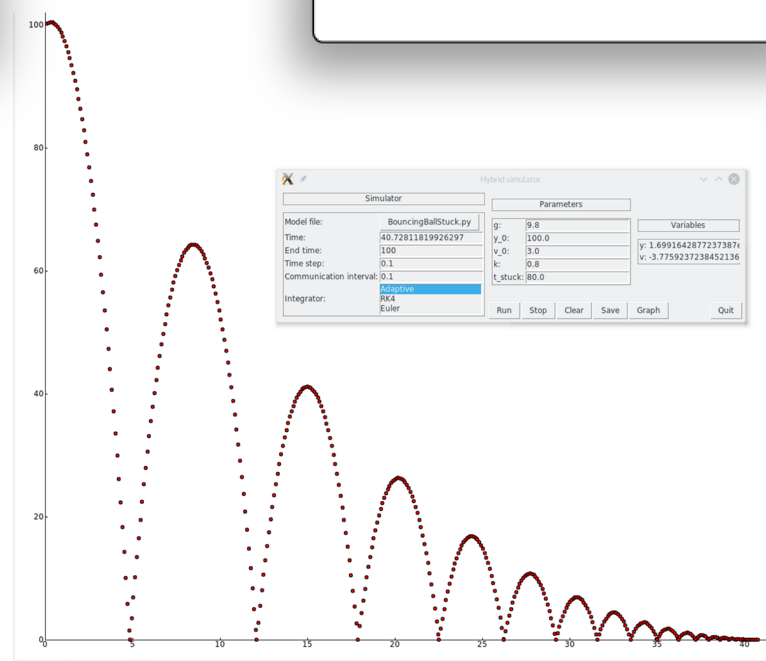
MODELICA

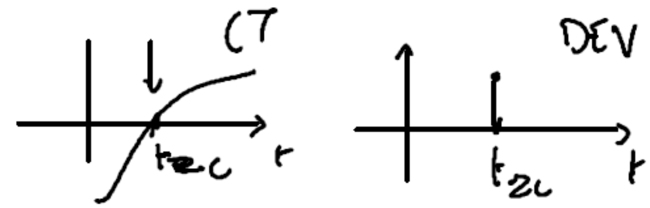
$$\begin{cases} \dot{y} = c_1 x + c_2 x^2 \\ \dot{z} = c_1 y & (\dot{z} = 0 \text{ WHEN NOT USED}) \\ \dot{w} = c_2 x \end{cases}$$

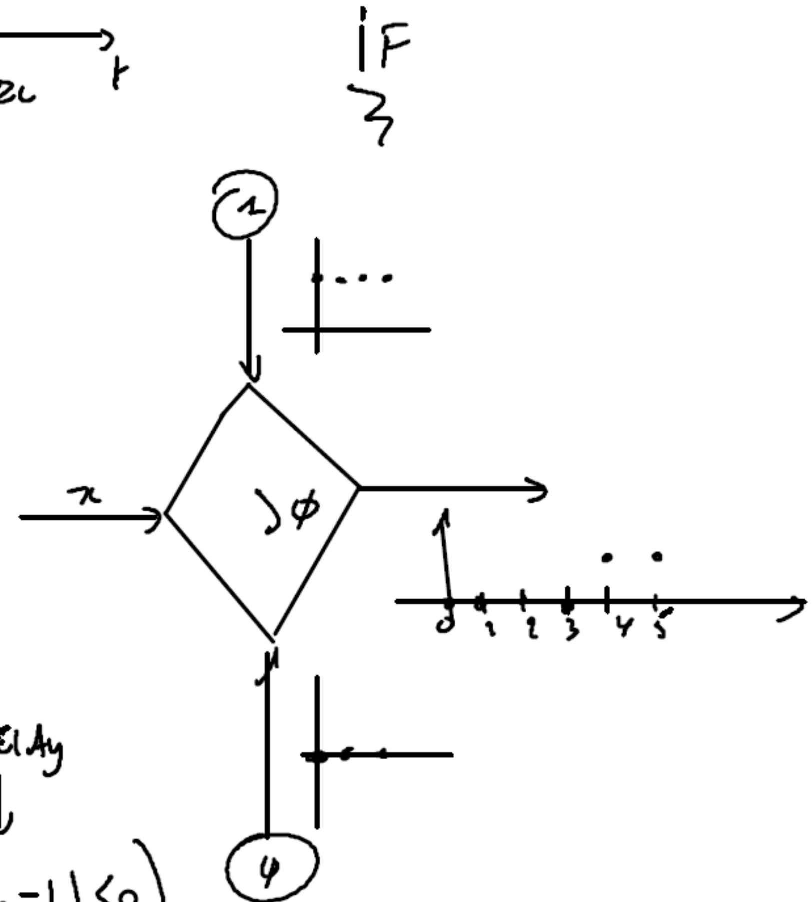
SO PIECEWISE CONTINUOUS




[when x +-] / k=-0.8; v_0=k*v; x_0=0
FreeFall
Stuck
Initial
startBouncing /
x_0=10; v_0=15
v_0
x_0
g
-9.81
I
C
v
x
[(when v-v_eps +-) &&
(x <= x_eps)] / x_0=0; v_0=0
0
kick / v_k=20; x_0 =x; v_0+=v_k

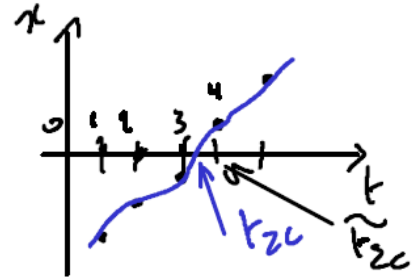

100
80
60
40
20
0
5
10
15
20
25
30
35
40
Hybrid simulator
Simulator
Model file: BouncingBallStuck.py
Time: 40.72811819926297
End time: 100
Time step: 0.1
Communication interval: 0.1
Integrator: Adaptive
RK4
Euler
Parameters
g: 9.8
y_0: 100.0
v_0: 3.0
k: 0.8
t_stuck: 80.0
Variables
y: 1.6991642877237387e
v: -3.7759237238452136
Run
Stop
Clear
Save
Graph
Quit
[BouncingBall] time vs y in red circles





WHEN: (CT → DT) → DEV SIGNAL

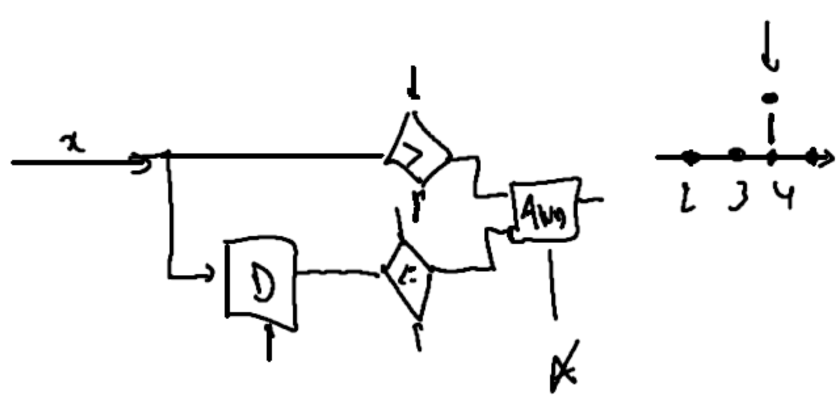
WHEN $(x(t) \neq 0)$



DISCR

DELAY

$$\tilde{t}_{zc}? : (\tilde{x}(\tilde{t}_{zc}) > 0) \& (\tilde{x}(\tilde{t}_{zc}-1) \leq 0)$$

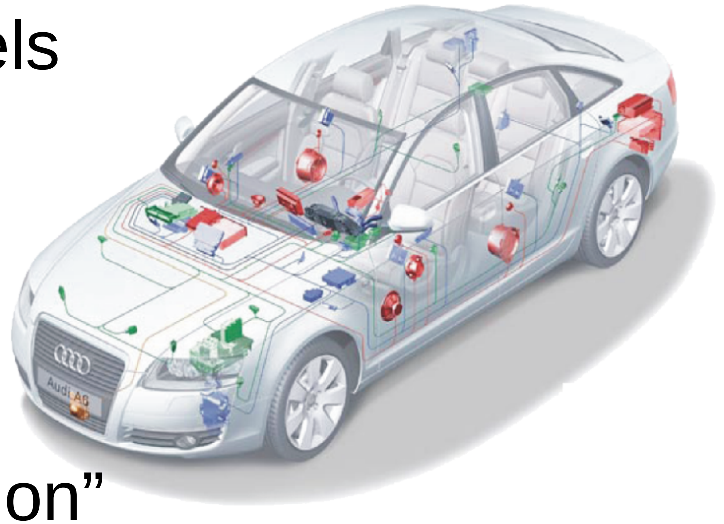


Physical Systems Modelling

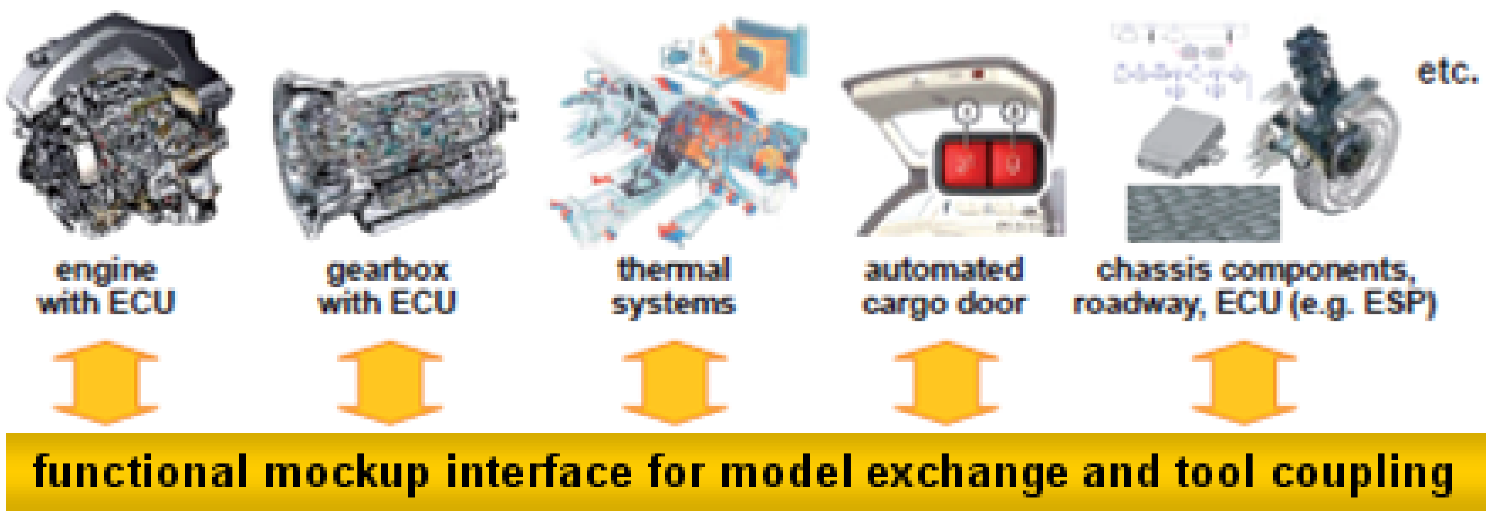
- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

Functional Mock-up Interface (FMI)

- **XML** + Binary Representation for Models
  - Standard
  - Modelling Tool Independent
  - +/- Black box …

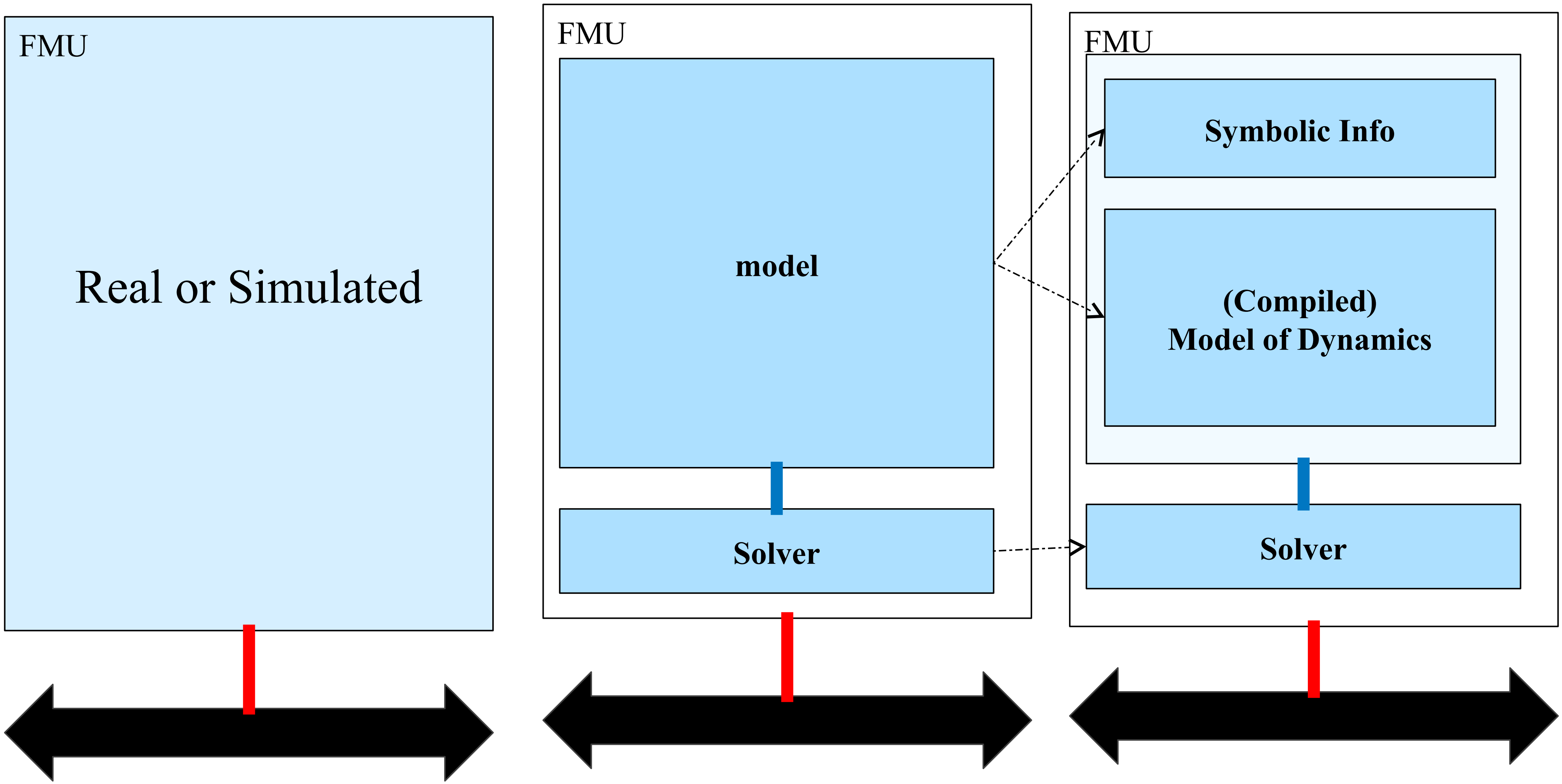


- Composed FMUs still need “orchestration”

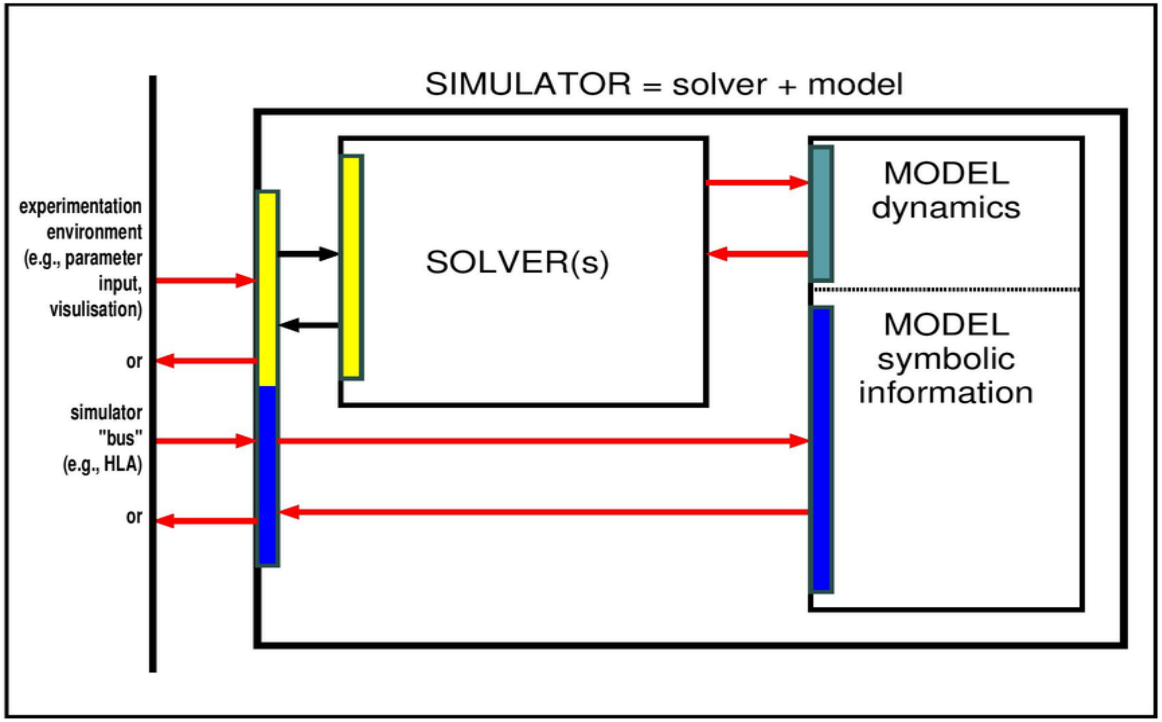


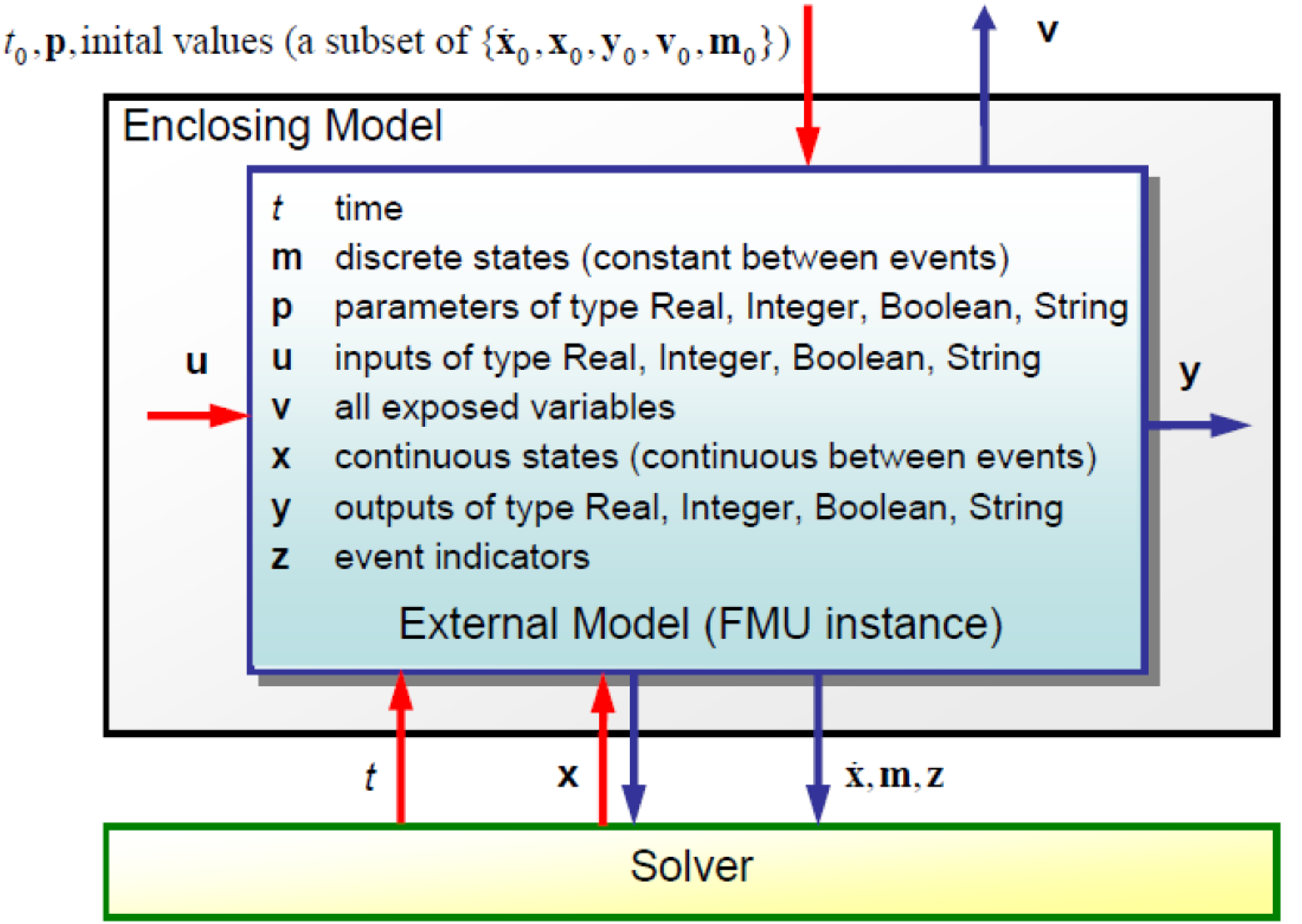
www.fmi-standard.org

Deconstructing an FMU

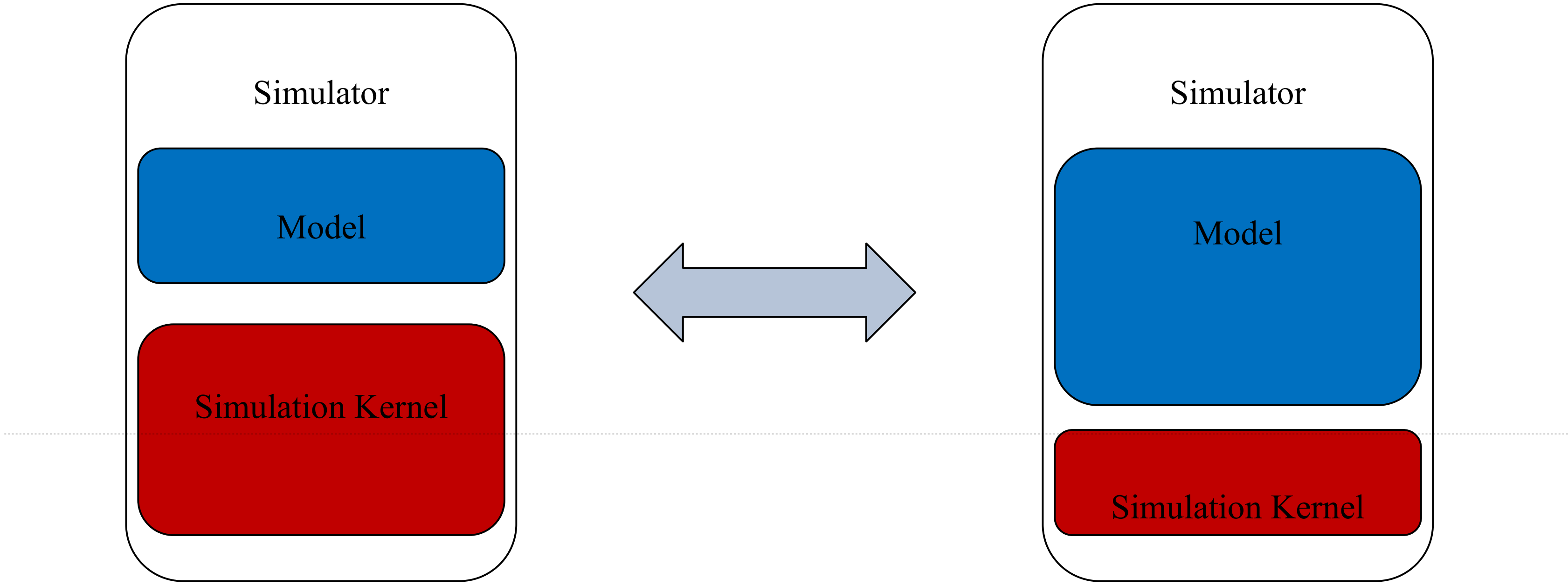


Model-Solver Interface
Simulator-Environment Interface




$t_0, \mathbf{p}$, inital values (a subset of $\{\dot{\mathbf{x}}_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{v}_0, \mathbf{m}_0\}$)
v
Enclosing Model
t time
m discrete states (constant between events)
p parameters of type Real, Integer, Boolean, String
u inputs of type Real, Integer, Boolean, String
v all exposed variables
x continuous states (continuous between events)
y outputs of type Real, Integer, Boolean, String
z event indicators
External Model (FMU instance)
u
y
t
x
$\dot{\mathbf{x}}, \mathbf{m}, \mathbf{z}$
Solver

meaningful operational semantics (Models of Computation)



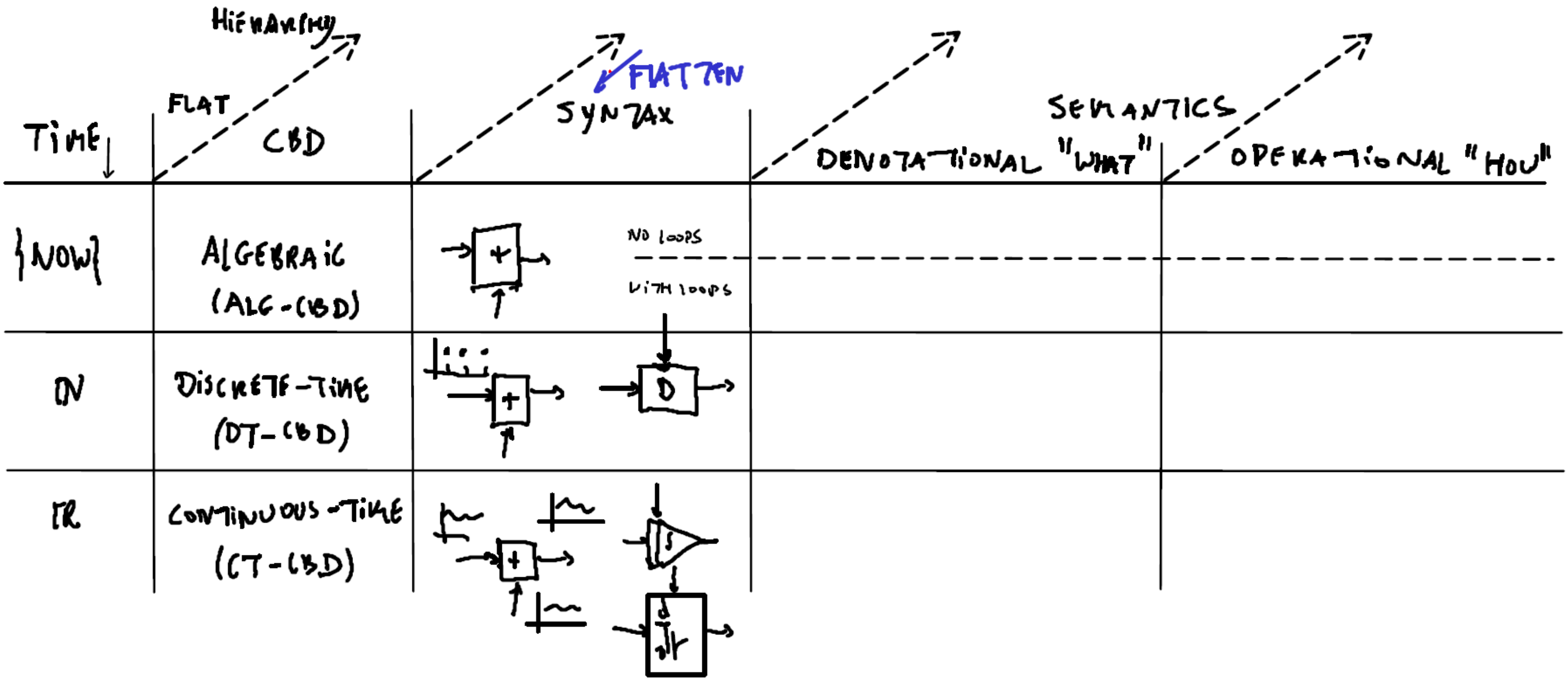
Traditional Simulator

Explicit Computational Semantics

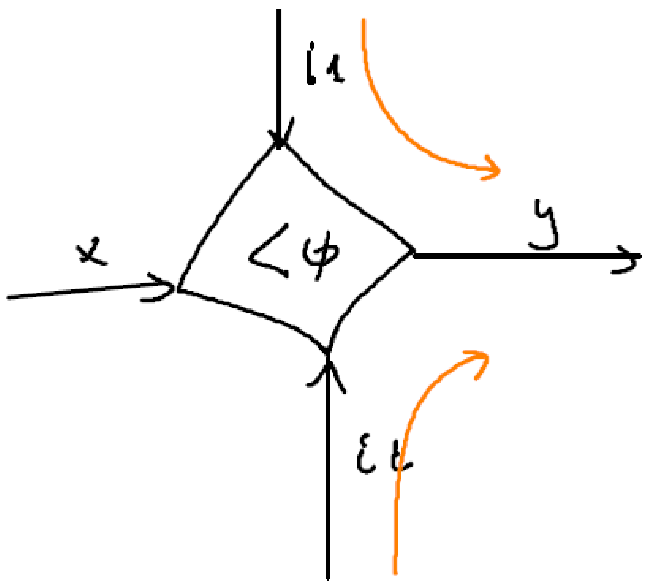
Physical Systems Modelling

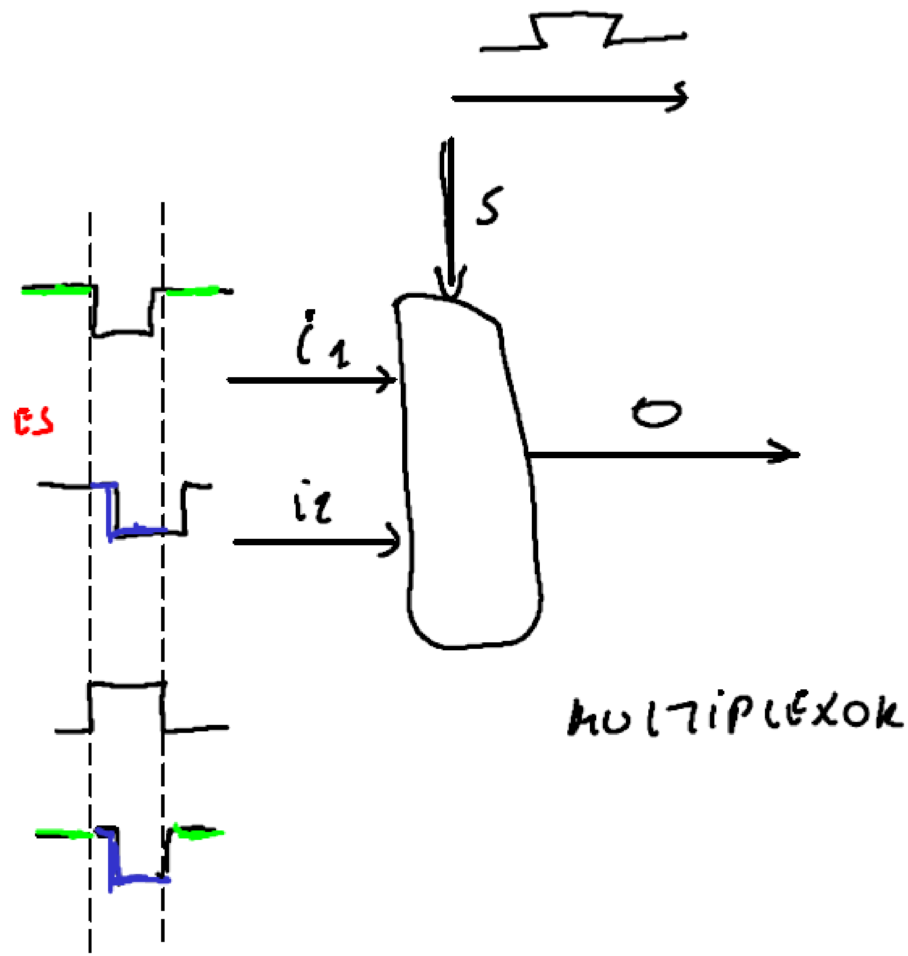
- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
  (Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
  (Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

++ Dynamic Structure

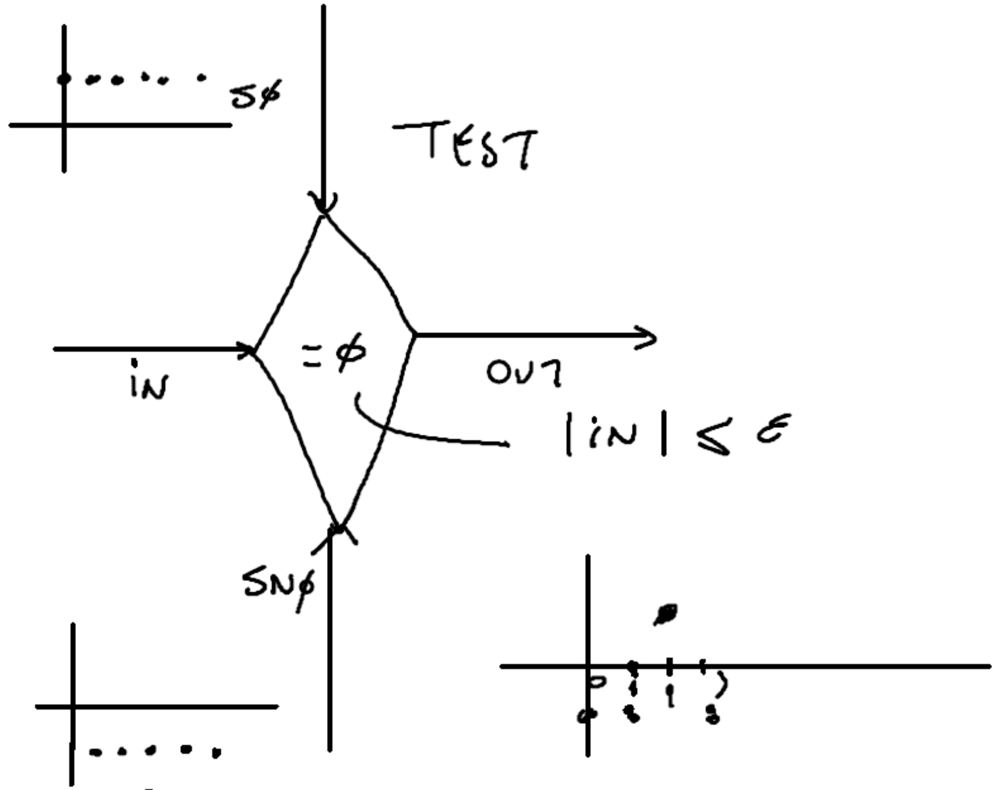


DYNAMIC STRUCTURE





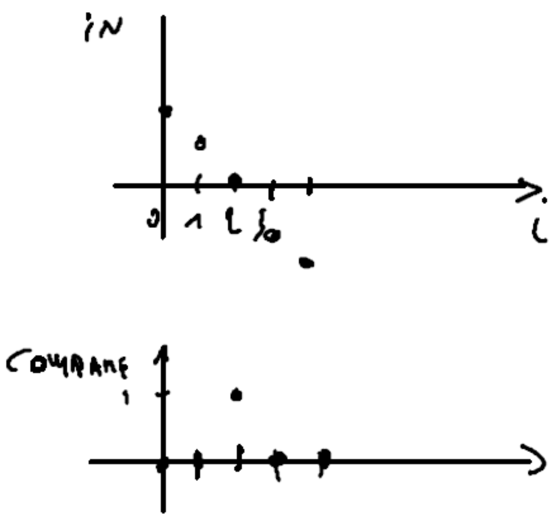
s
i_1
ES
i_2
o
MULTIPLEXOR

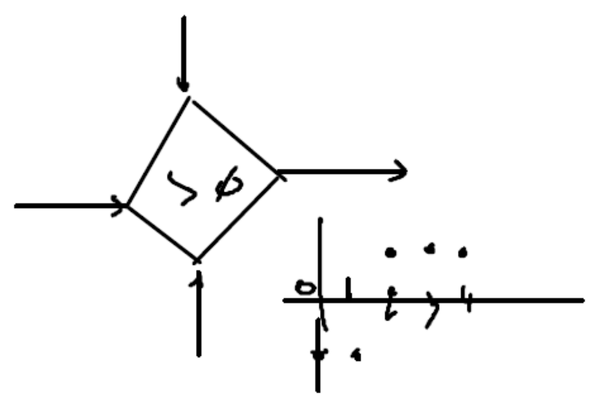


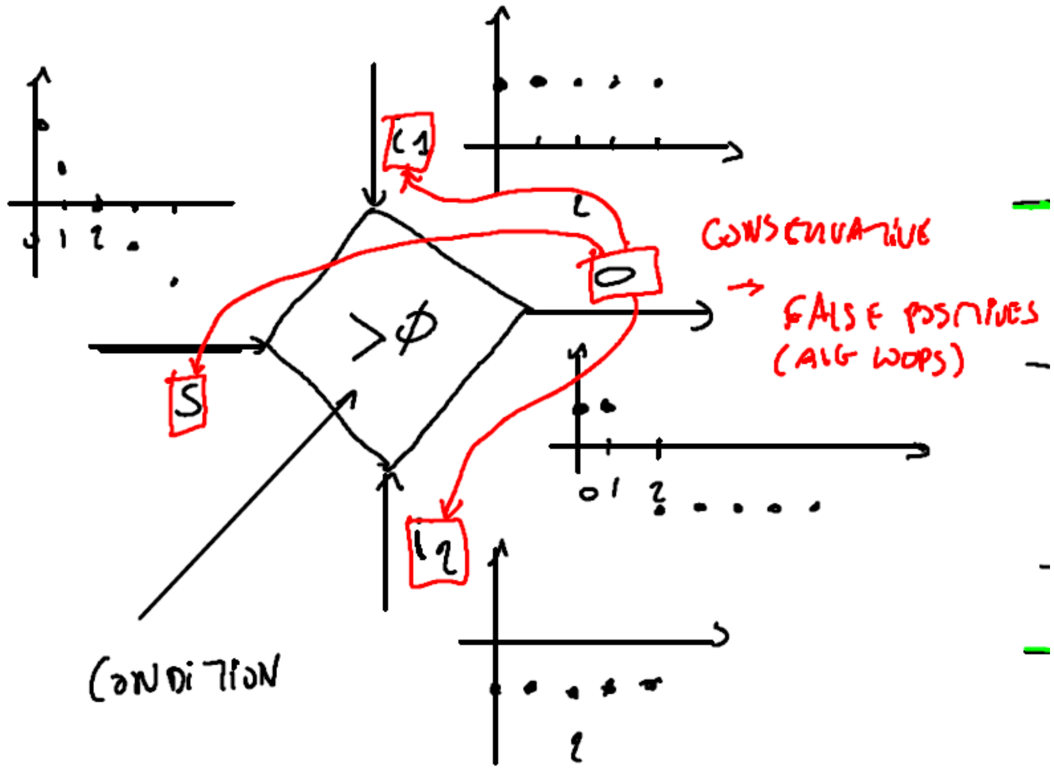
$\mathbb{N}, \mathbb{Z}$ init $\quad i_1 == i_2$

$\mathbb{R}$ float, double $\quad r_1 == r_2$

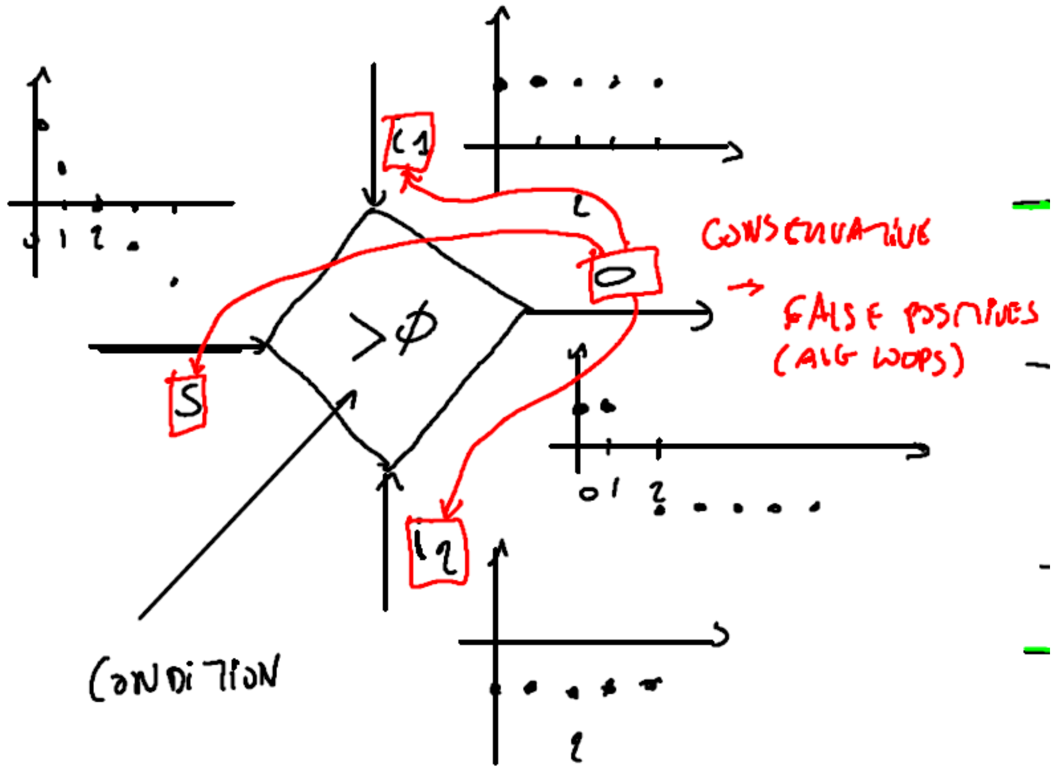
$|r_1 - r_2| \leq \varepsilon$

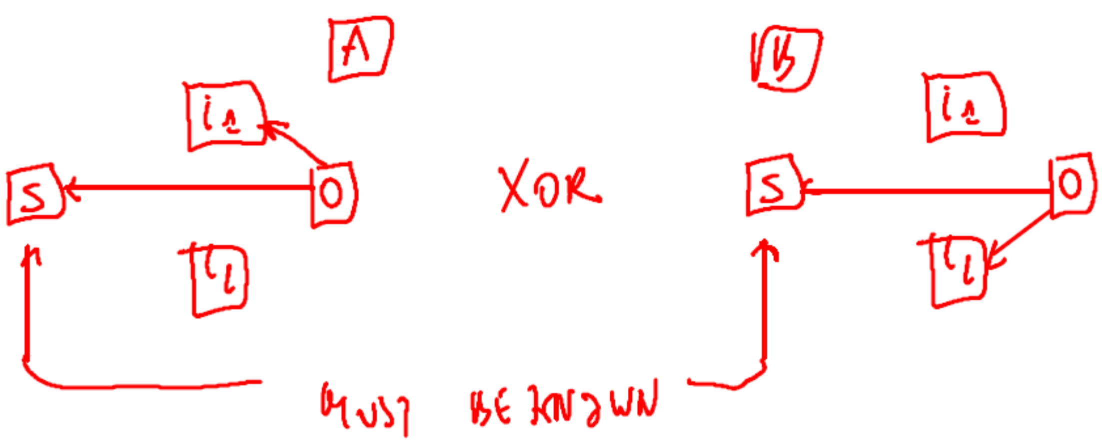






>Ø
S
CONDITION
CONSERVATIVE
→ FALSE POSITIVES
(AVG LOOPS)

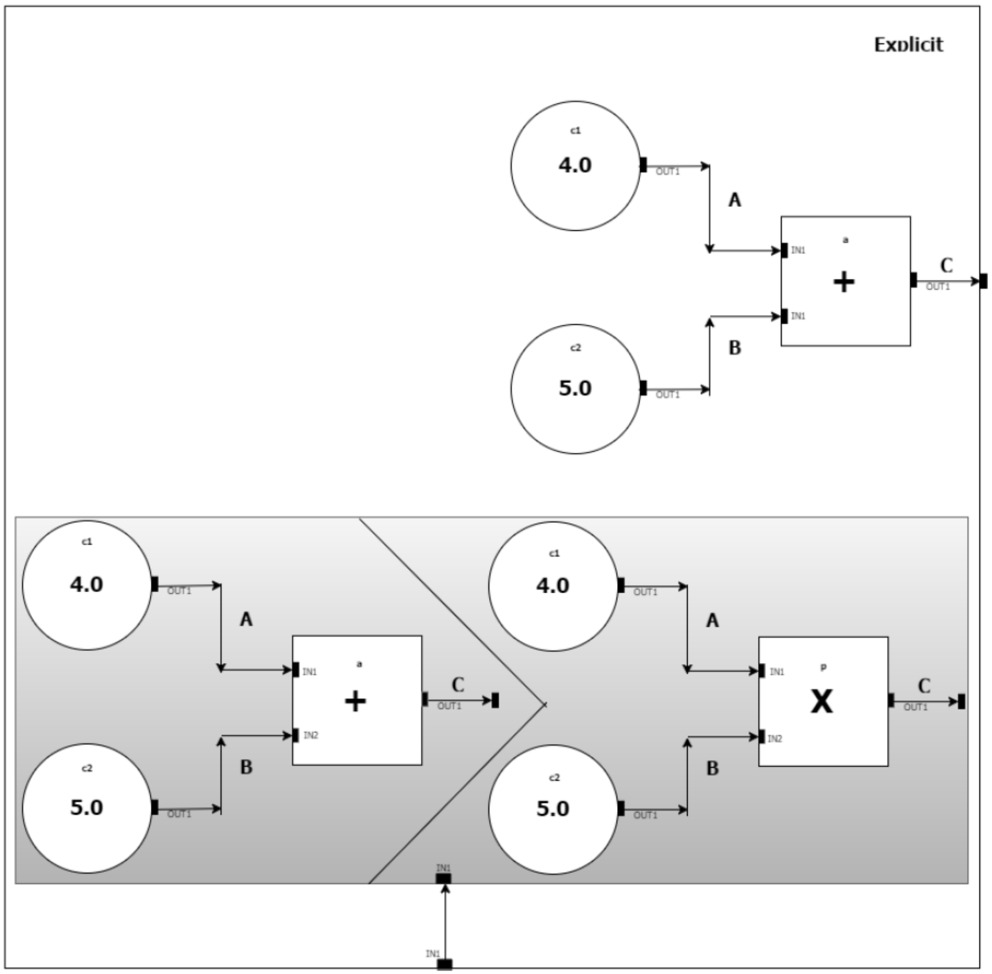


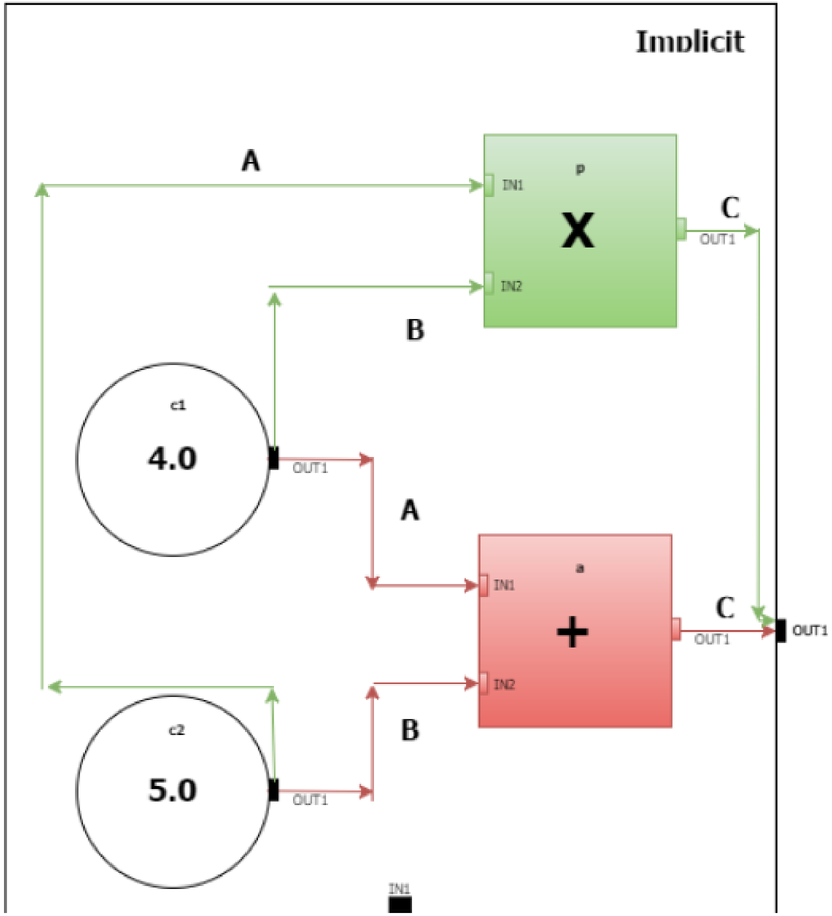


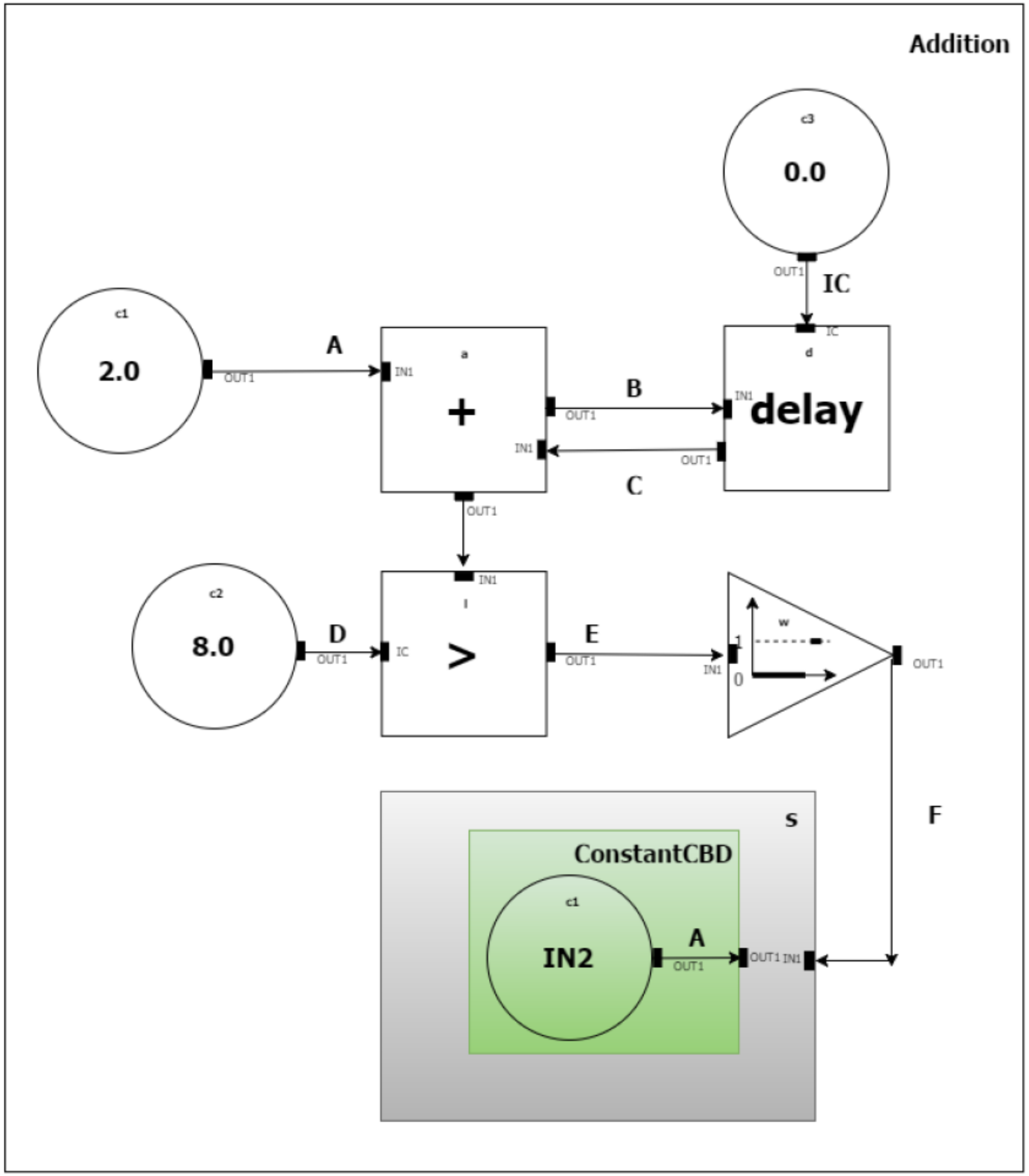
1. BUILD DEP. GRAPH UPTO [S]
2. PARTIAL SORT/LOOP DET UPTO [S]
3. COMPLETE DEP GRAPH WITH [A] OR [B]
4. FULL SORT/LOOP DETECT

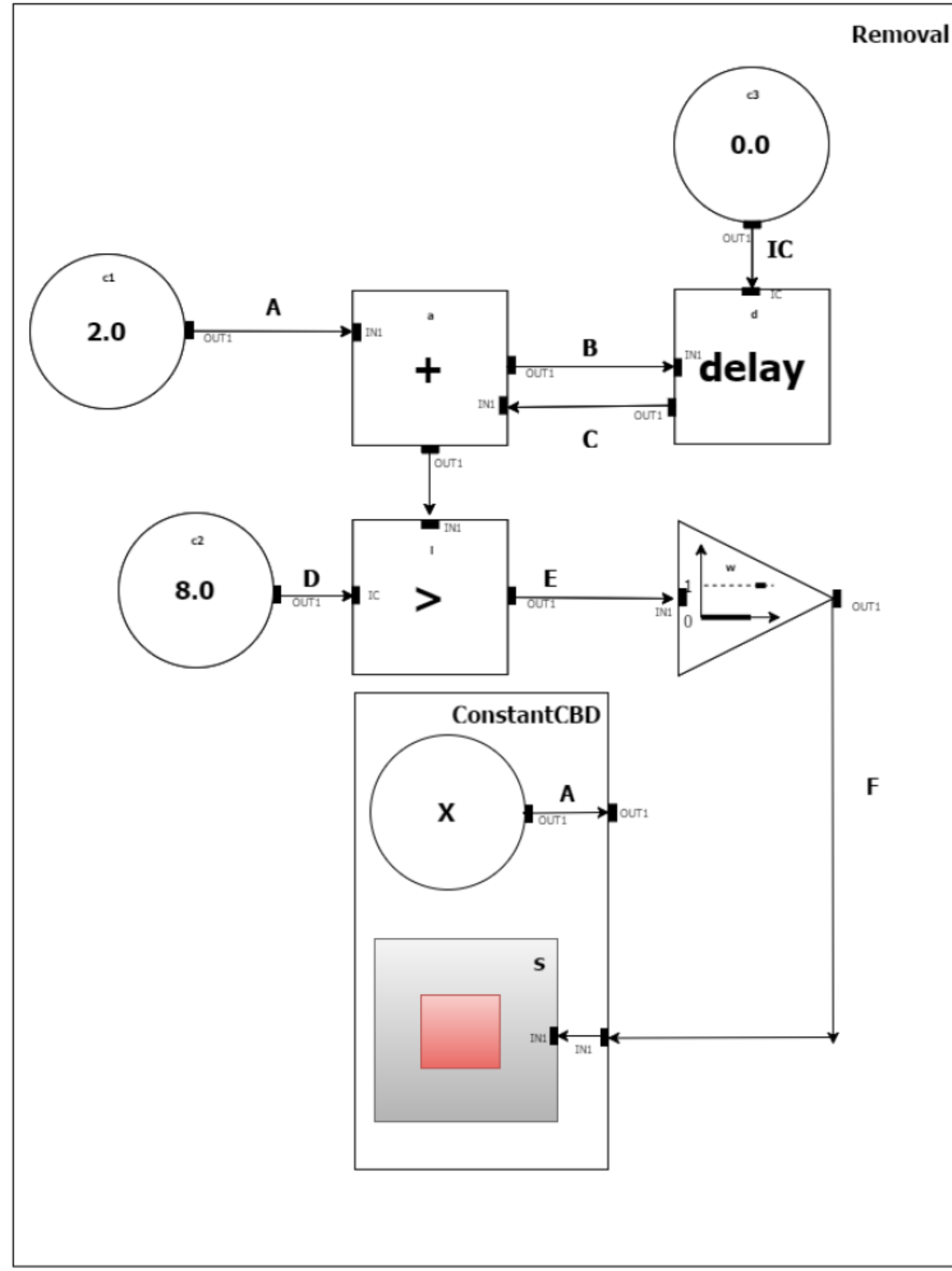
MULTIPLE TEST BLOCK

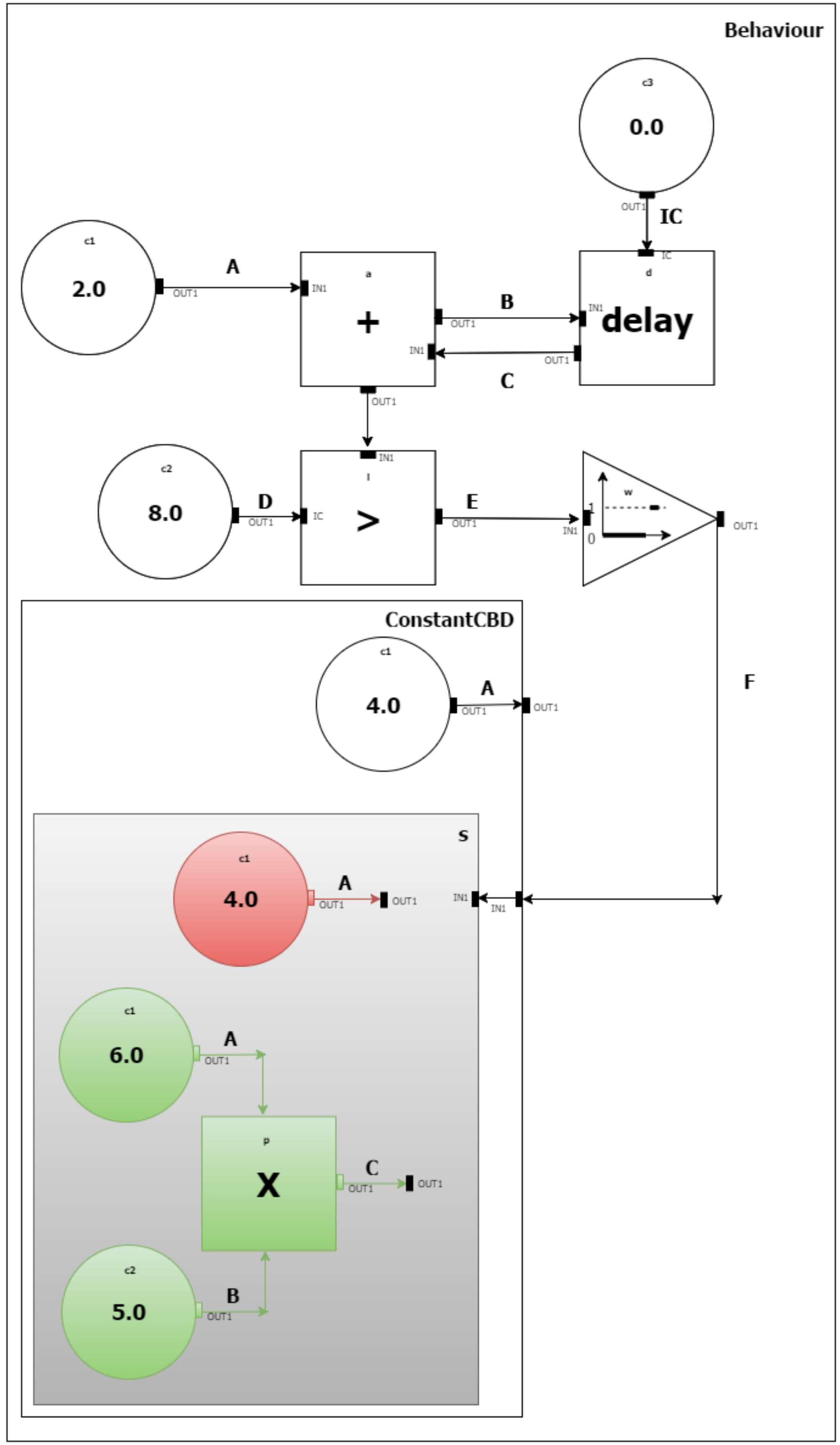
FIXED POINT


Explicit
c1
4.0
OUT1
A
a
+
IN1
IN1
C
OUT1
c2
5.0
OUT1
B
p
X
IN2
IN1

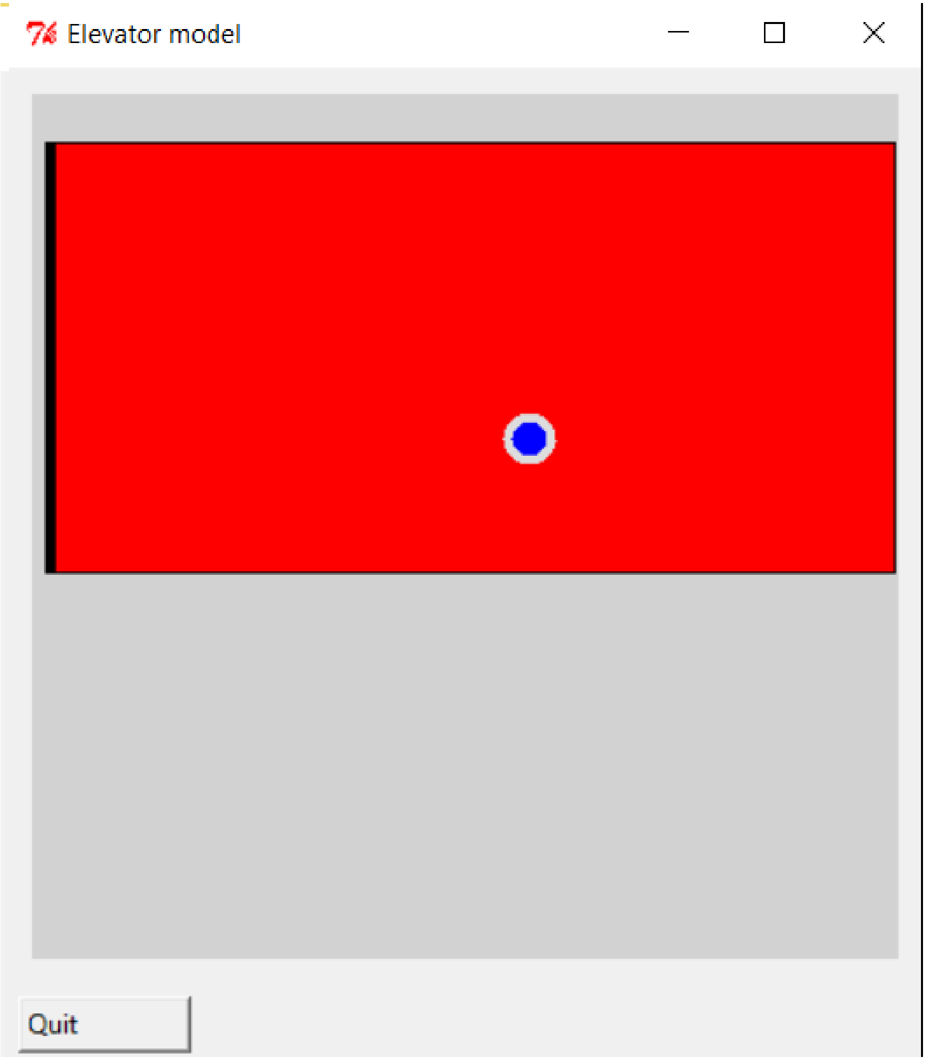

Implicit
A
p
X
IN1
IN2
C
OUT1
B
c1
4.0
OUT1
A
a
+
IN1
IN2
C
OUT1
OUT1
B
c2
5.0
OUT1
IN1


Addition
c3
0.0
OUT1
IC
c1
2.0
OUT1
A
a
IN1
+
OUT1
IN1
B
C
d
IC
IN1
delay
OUT1
OUT1
c2
8.0
OUT1
D
IN1
IC
>
OUT1
E
IN1
w
1
0
OUT1
F
s
ConstantCBD
c1
IN2
A
OUT1
OUT1
IN1

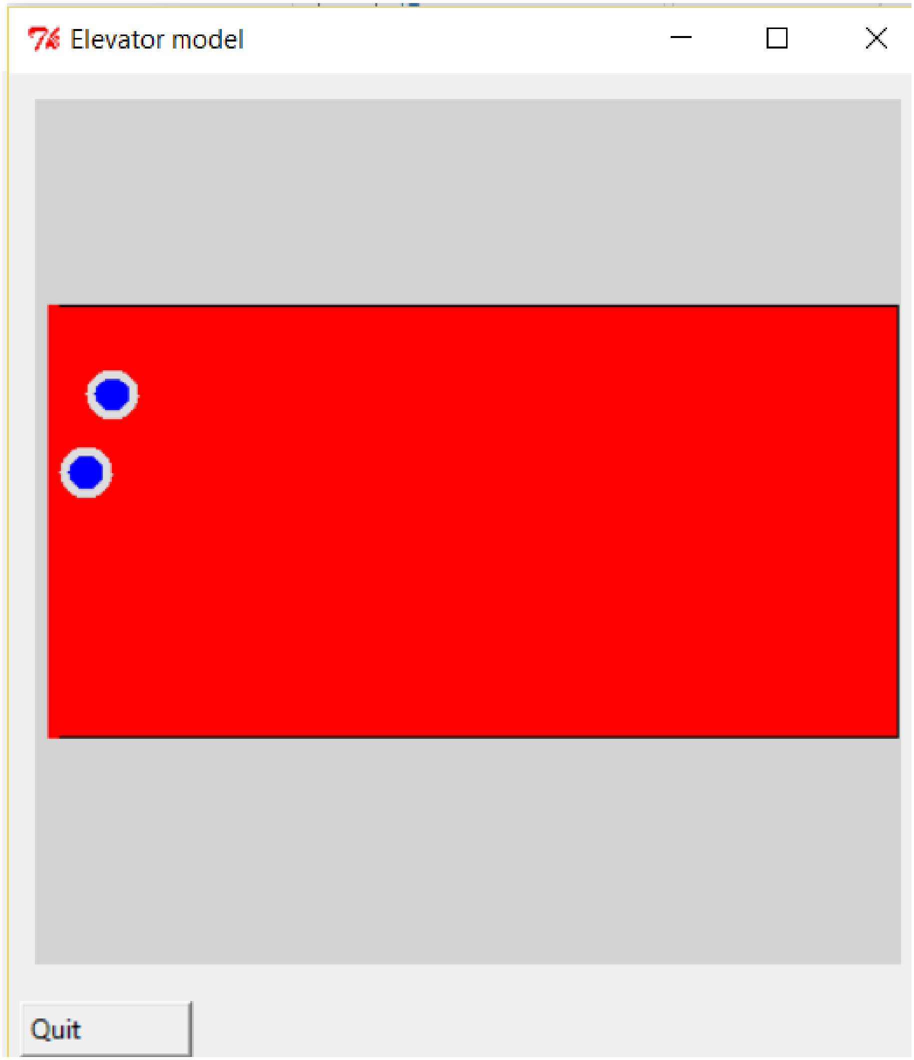

Removal
c3
0.0
OUT1
IC
c1
2.0
OUT1
A
a
IN1
+
OUT1
IN1
B
C
d
IC
IN1
delay
OUT1
OUT1
c2
8.0
D
OUT1
IN1
IC
>
OUT1
E
IN1
w
1
0
OUT1
ConstantCBD
X
A
OUT1
OUT1
F
s
IN1
IN1



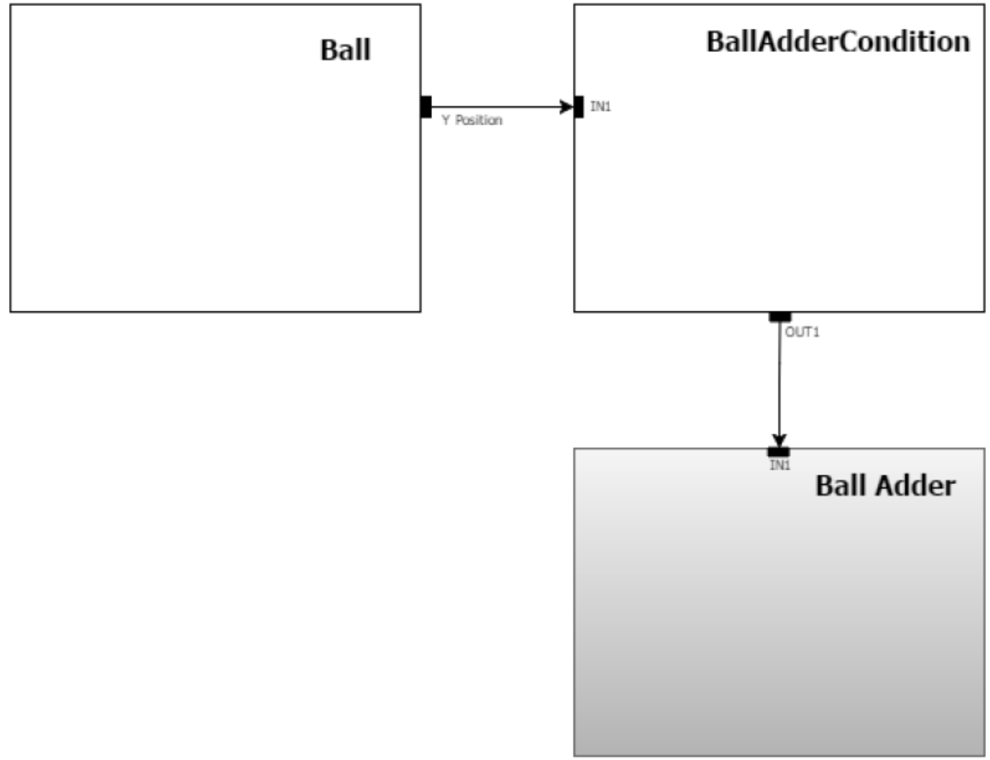
Behaviour
c3
0.0
OUT1
IC
c1
2.0
OUT1
A
IN1
a
+
B
OUT1
IN1
d
delay
IN1
OUT1
C
OUT1
IN1
c2
8.0
D
OUT1
IC
l
>
E
OUT1
IN1
w
1
0
OUT1
F
ConstantCBD
c1
4.0
A
OUT1
OUT1
s
c1
4.0
A
OUT1
OUT1
IN1
IN1
c1
6.0
A
OUT1
p
X
C
OUT1
OUT1
c2
5.0
B
OUT1

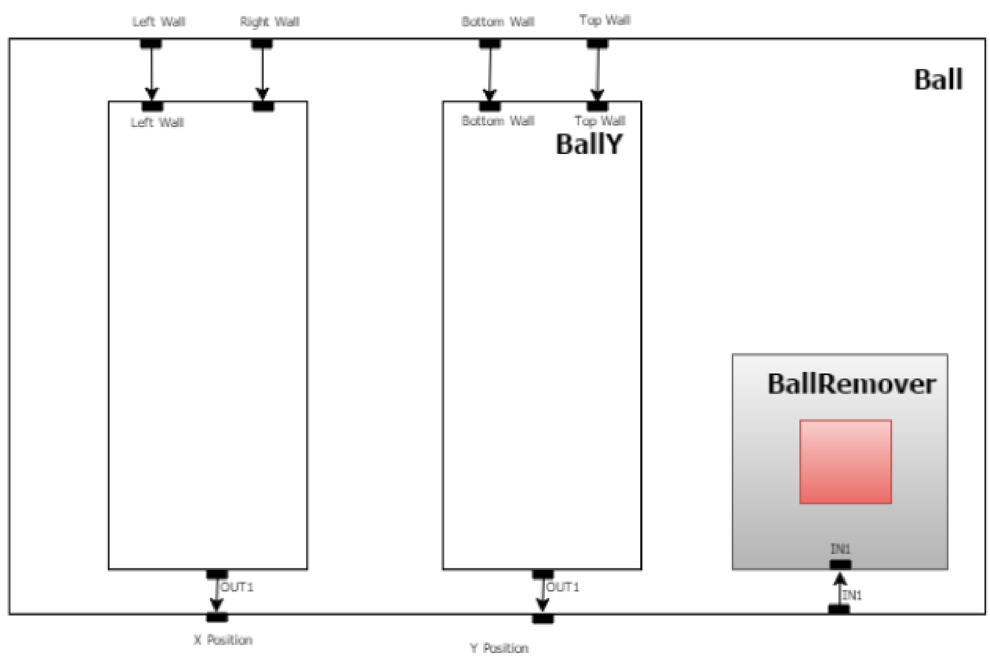


Elevator model
Quit



Elevator model
Quit


Ball
Y Position
BallAdderCondition
IN1
OUT1
IN1
Ball Adder


Left Wall
Right Wall
Bottom Wall
Top Wall
Ball
Left Wall
Bottom Wall
Top Wall
BallY
BallRemover
IN1
OUT1
OUT1
IN1
X Position
Y Position

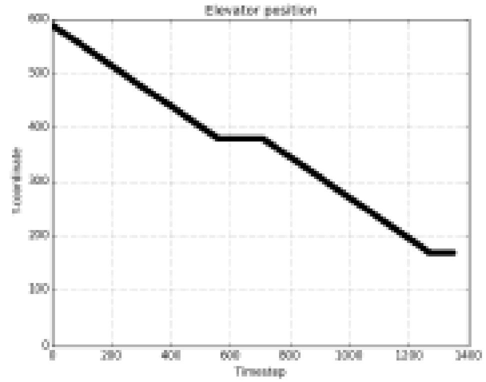


Figure 4.10: Y-position of the elevator

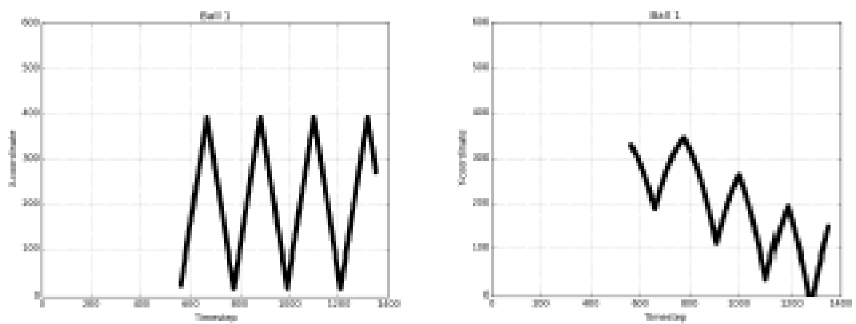


Figure 4.11: Position of ball 1

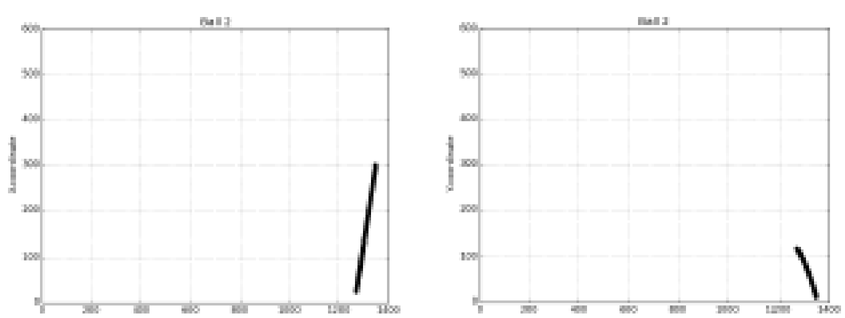


Figure 4.12: Position of ball 2