RECENT DEVELOPMENTS FOR MONO- OR MULTI-RATE PARALLEL REAL-TIME CO-SIMULATION

EXTRAPOLATION AND SCHEDULING FOR MULTICORE ARCHITECTURES

Abir El Feki, Salah Eddine Saidi, Nicolas Pernet, Laurent Duval, Mongi Ben Gaid (IFPEN)

Yves Sorel, Daniel Simon (INRIA)

Cyril Faure (CEA)

OUTLINE

- Background
  - **Co-simulation: context & challenges**
  - Real-time Co-Simulation: an open problem
- Improving parallelism with the RCosim approach: Refined Co-simulation
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- Mapping real-time constraints for HiL
- Future work

BACKGROUND

- Co-simulation: Alternative to monolithic simulation → Simulation of a complex system using several coupled subsystems
  - A subsystem is modeled using the most appropriate tool instead of using a single modeling software
  - Subsystems are modeled and run in a segregated manner → The equations of each model are integrated using a solver separately
  - During the execution models exchange data → A synchronization mechanism is used between the models, in such a way that models update their inputs and outputs according to assigned communication steps
  - Easy upgrade, reuse, and exchange of models

BACKGROUND

- Co-simulation: Alternative to monolithic simulation → Simulation of a complex system using several coupled subsystems
  - A subsystem is modeled using the most appropriate tool instead of using a single modeling software
  - Subsystems are modeled and run in a segregated manner → The equations of each model are integrated using a solver separately
  - During the execution models exchange data → A synchronization mechanism is used between the models, in such a way that models update their inputs and outputs according to assigned communication steps
  - Easy upgrade, reuse, and exchange of models
  - Heterogeneous ODE models → Time consuming simulations

Model 1 + Model 2 + Model 3 + Model 4 = Complex model → Time consuming simulation

BACKGROUND (CONT'D)

- A multi-core co-simulation kernel: Why?
  - System-level simulation leads to agglomerate models which are classically disconnected, increasing the CPU demand at simulation time
  - Simulation time becomes more and more a metric for model complexity
  - Most 0D/1D simulation tools have mono-core kernel while mono-core computers are endangered

How long will this CPU power remain unused ?

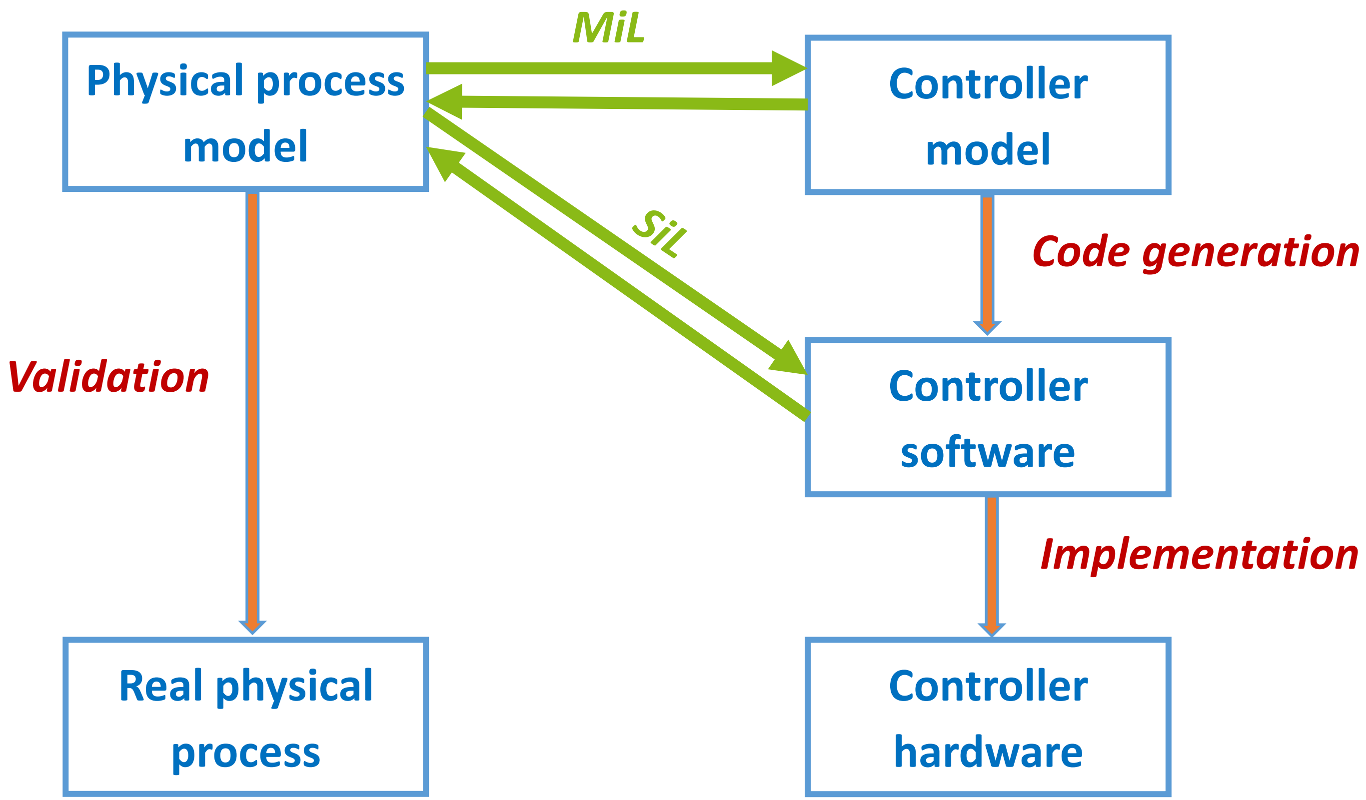
INSTITUT CARNOT IFPEN Transports Energie

IFP Energies nouvelles

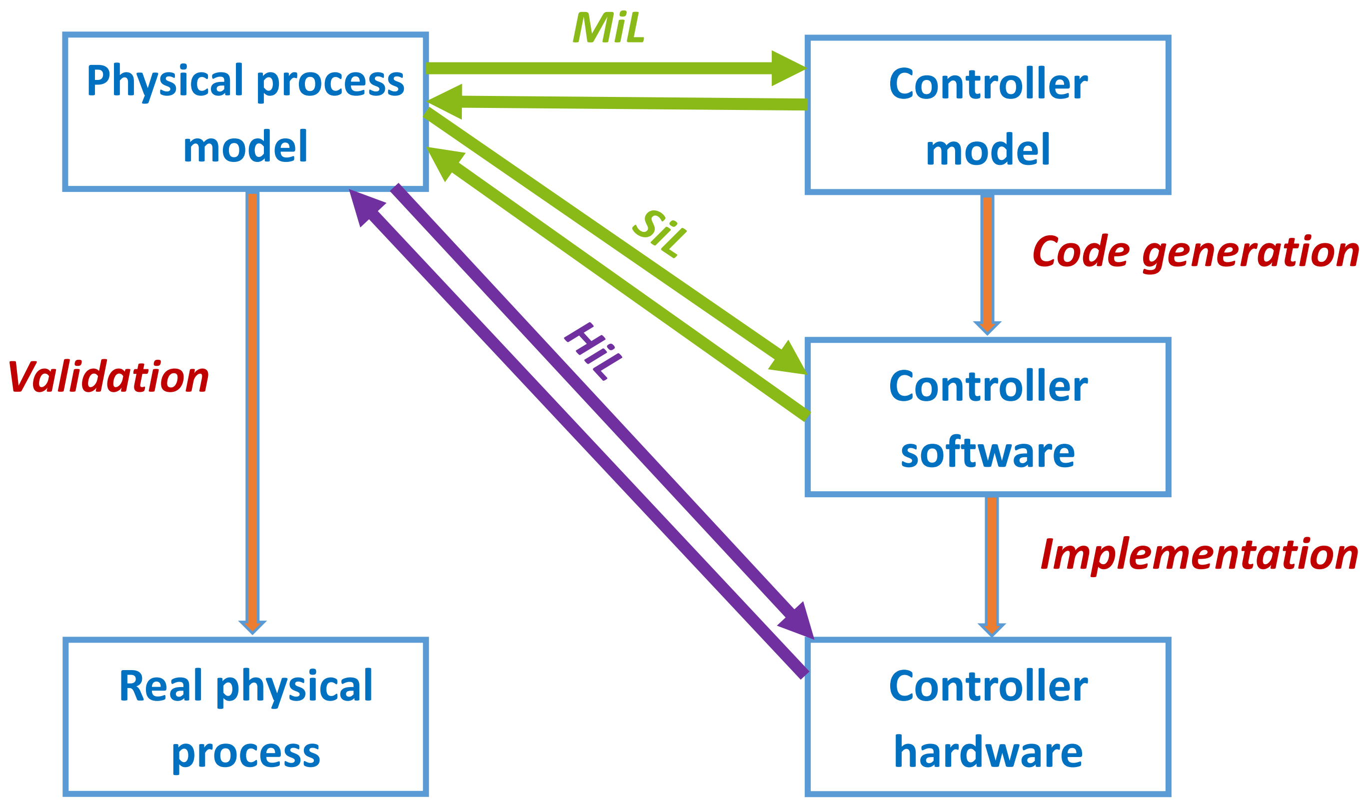
OUTLINE

- Background
  - Co-simulation: context & challenges
  - **Real-time Co-Simulation: an open problem**
- Improving parallelism with the RCosim approach: Refined Co-simulation
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- Mapping real-time constraints for HiL
- Future work

REAL-TIME SIMULATION
NEEDS FOR CPS VALIDATION

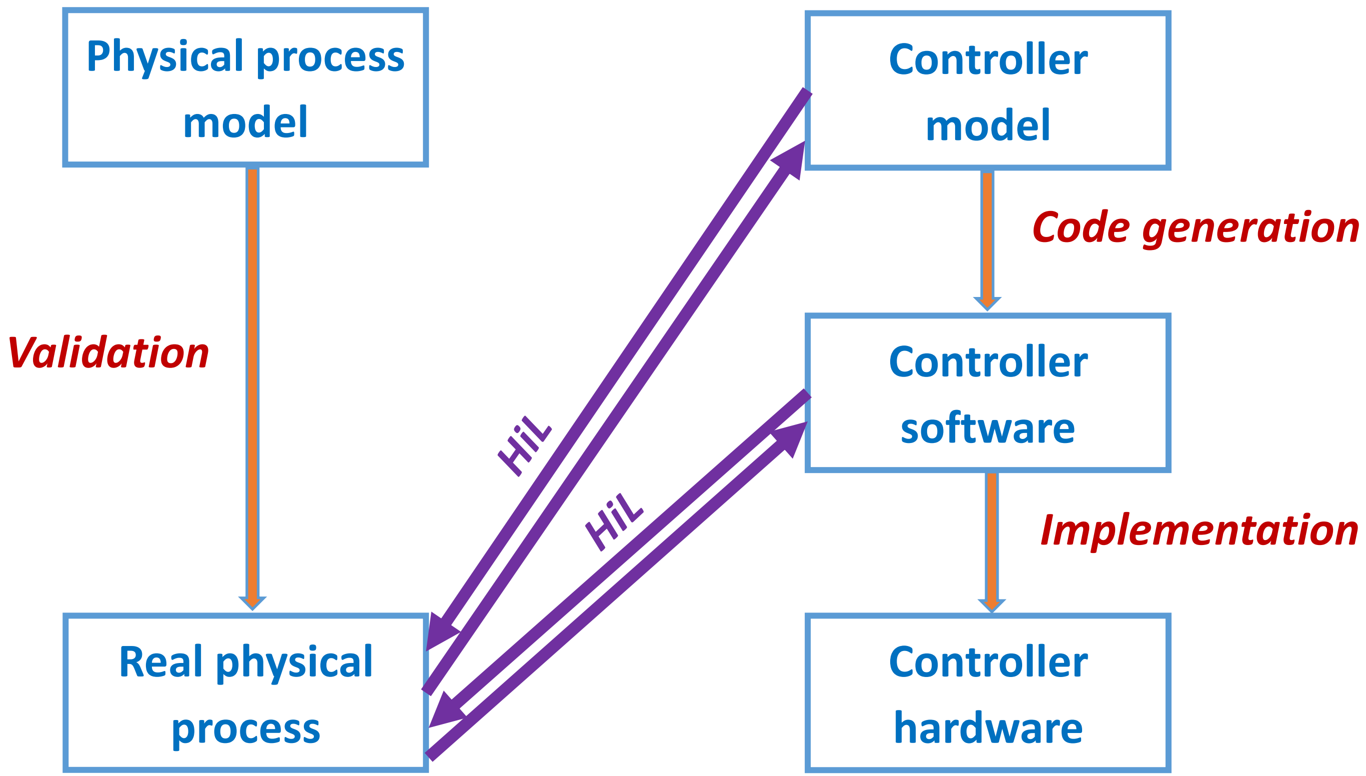


REAL-TIME SIMULATION
NEEDS FOR CPS VALIDATION



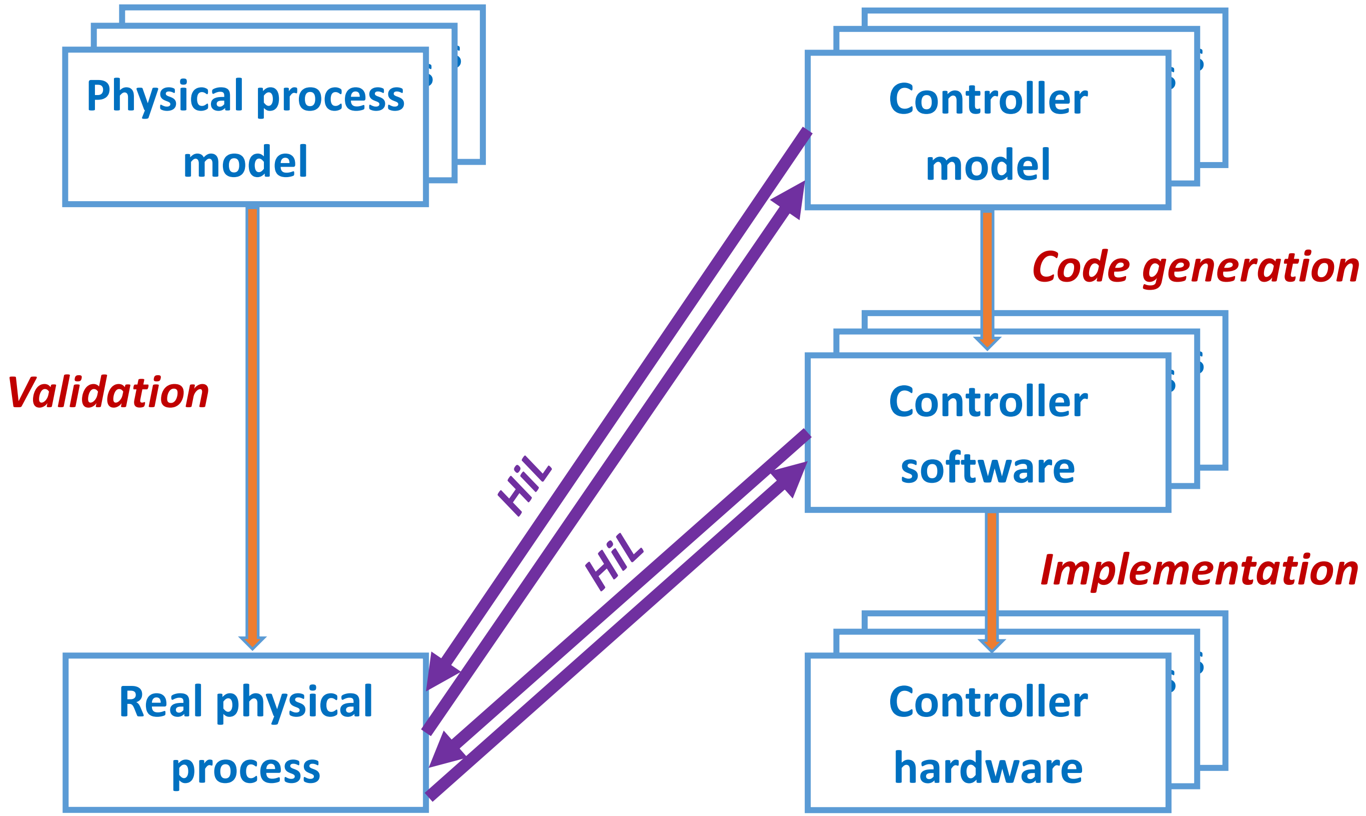
Hardware-in-the-Loop ➔ real-time constraints

REAL-TIME SIMULATION
NEEDS FOR CPS VALIDATION



Hardware-in-the-Loop ➔ real-time constraints

REAL-TIME SIMULATION
NEEDS FOR CPS VALIDATION



Hardware-in-the-Loop ➔ real-time constraints

OUTLINE

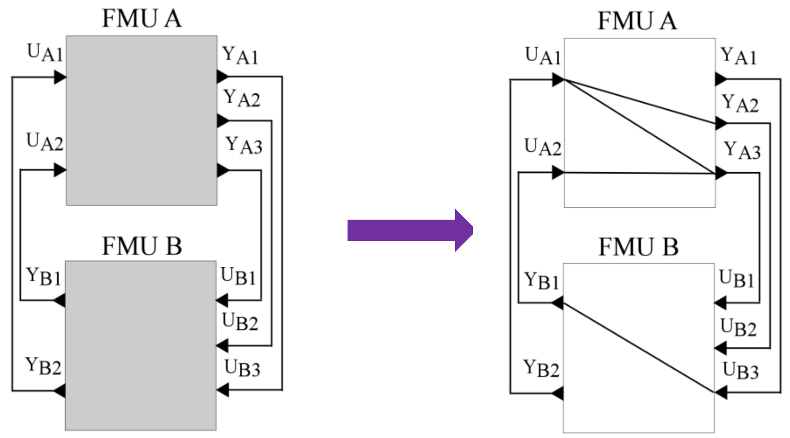
- Background
  - Co-simulation: context & challenges
  - Real-time Co-Simulation: an open problem
- **Improving parallelism with the RCosim approach: Refined Co-simulation**
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- Mapping real-time constraints for HiL
- Summary and outlook

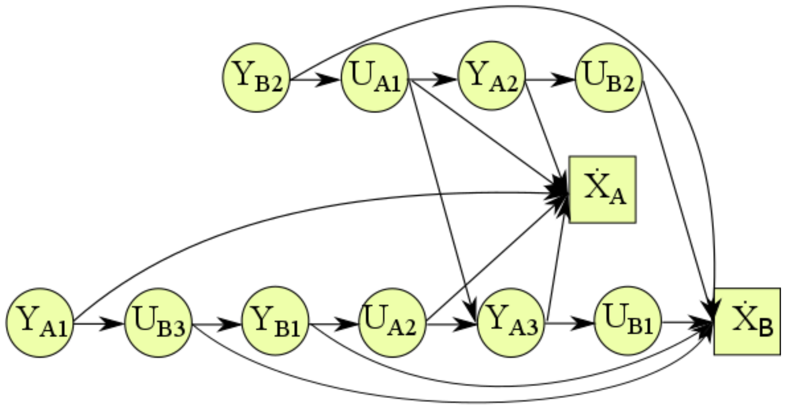
INSTITUT CARNOT IFPEN Transports Energie

Energies nouvelles

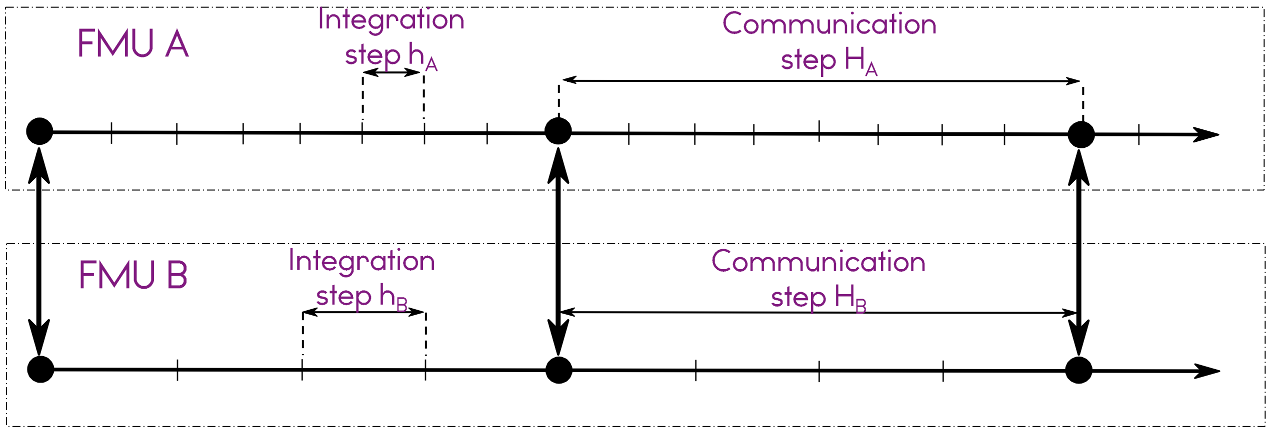
RCOSIM: REFINED CO-SIMULATION
DATAFLOW GRAPH OF FMUS

- Inter FMU dependencies specified by the user
- Identify locally if Y is dependent on U or not
  - FMI gives relationships between each Y and U
  - With FMI each I/O is computed with a different FMU functions
- Build refined dependency graph
  - Vertices: operations, a set of FMU functions
    - updateOut, updateIn, and updateState
  - Directed edges: precedencies between operations
  - Ordinary Differential Equations (ODEs)
    - No algebraic loops
    - Directed Acyclic Graph (DAG)
- Apply a multi-core scheduling heuristic on the dataflow graph

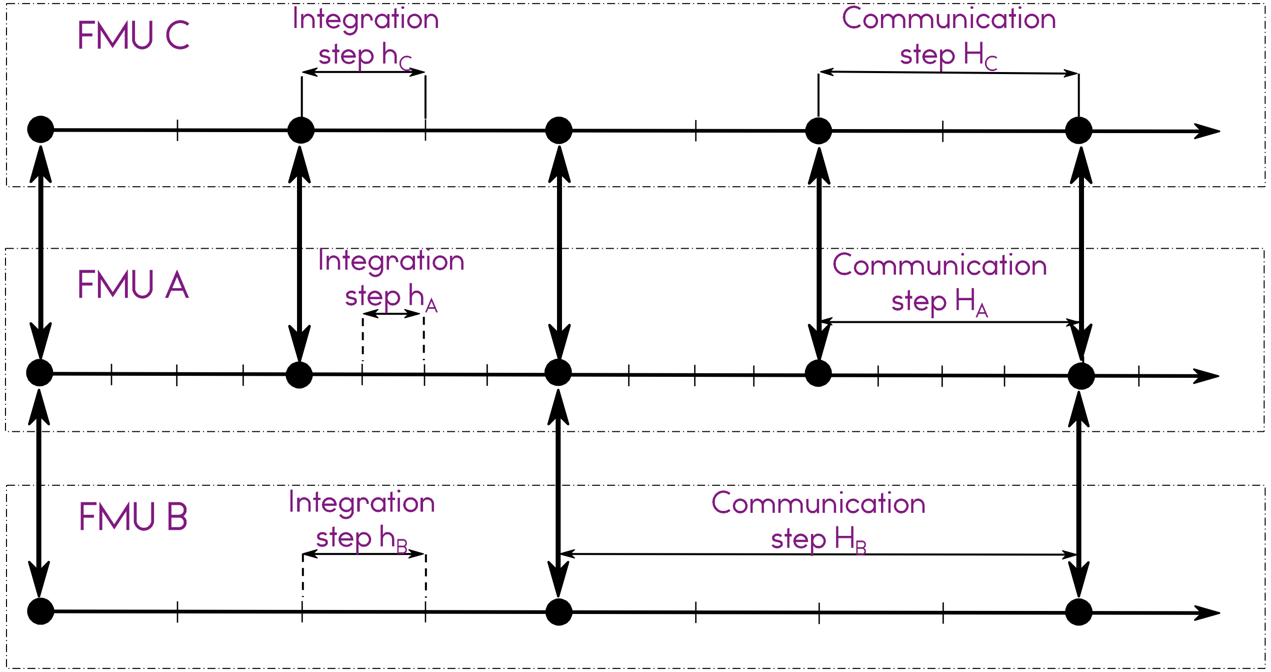




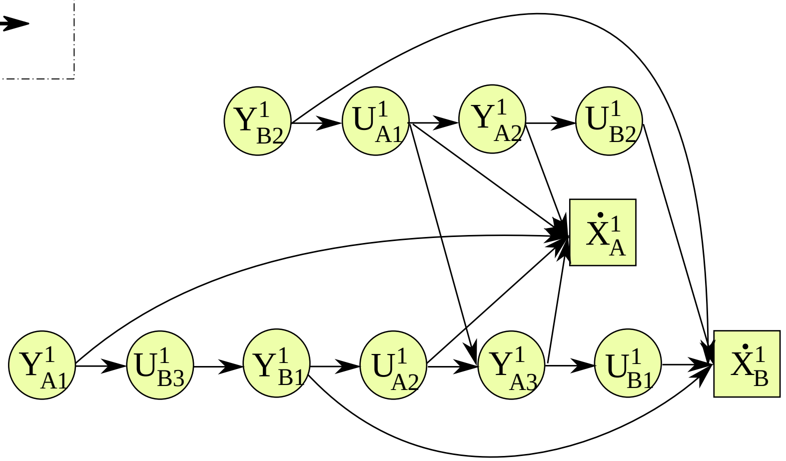
MUO-RCOSIM
EXTEND RCOSIM TO HANDLE MULTI-RATE CO-SIMULATION



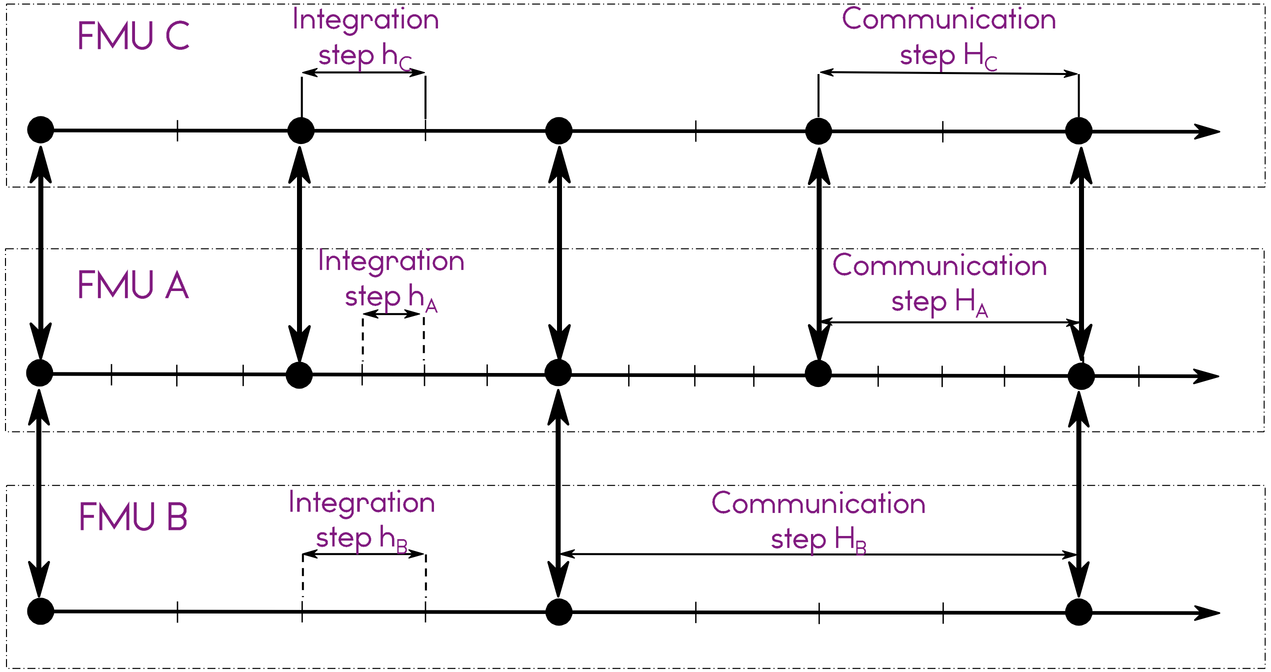
MUO-RCOSIM
EXTEND RCOSIM TO HANDLE MULTI-RATE CO-SIMULATION



- Multi-rate co-simulation
  - Update the I/O of each FMU according to its communication step
- Need for a repeatable pattern of the multi-rate graph execution
- Repeat each operation r_i= HS/H(o_i) times, HS=lcm (H(o_1), H(o_2), …, H(o_n))
- E.g: $H_B = 2 \times H_A$

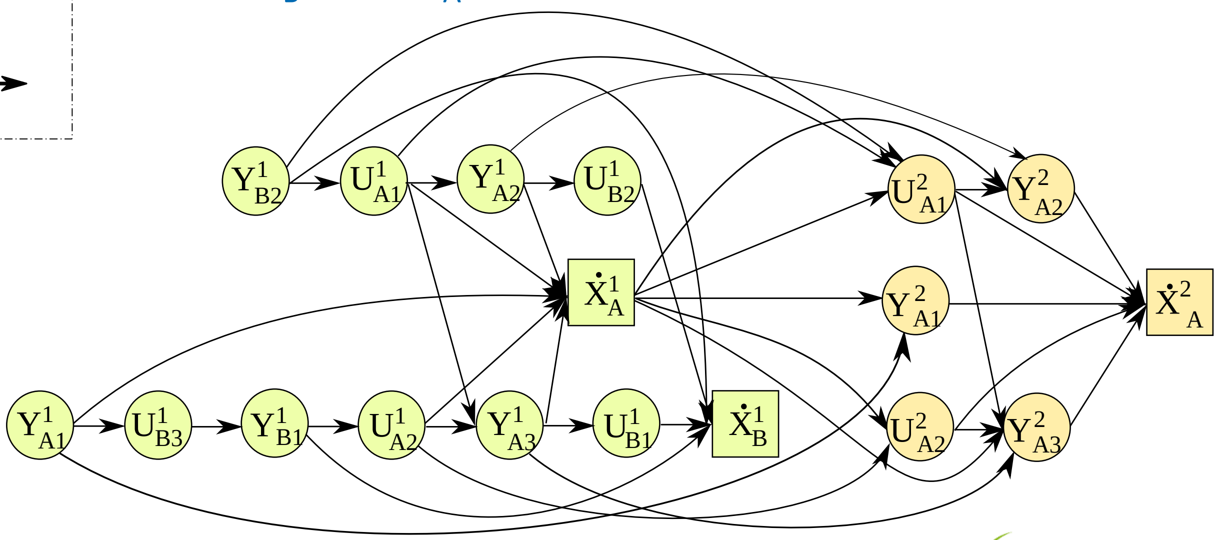


MUO-RCOSIM
EXTEND RCOSIM TO HANDLE MULTI-RATE CO-SIMULATION



Type of dependency	Slow to Fast	Fast to Slow
Communication steps	$H(o_i) > H(o_j)$	$H(o_i) < H(o_j)$
Edge creation rule	$o_i^s \rightarrow o_j^u$, $s \in \{1, 2, \dots, r_i\}$ $u = \left\lceil s \times \frac{H(o_i)}{H(o_j)} \right\rceil$	$o_i^s \rightarrow o_j^u$, $u \in \{1, 2, \dots, r_j\}$ $s = \left\lfloor u \times \frac{H(o_j)}{H(o_i)} \right\rfloor$

- Multi-rate co-simulation
  - Update the I/O of each FMU according to its communication step
- Need for a repeatable pattern of the multi-rate graph execution
- Repeat each operation $r_i = HS/H(o_i)$ times, HS=lcm $(H(o_1), H(o_2), \dots, H(o_n))$
- E.g: $H_B = 2 \times H_A$



MUO-RCOSIM
MULTI-RATE GRAPH TRANSFORMATION (CONT'D)

Multi-Rate Graph Transformation Algorithm

1) Compute the hyper-step HS=lcm ($H(o_1)$, $H(o_2)$, …, $H(o_n)$)
2) For each operation o_i in the graph
   - Compute the repetition factor r_i= HS/ H(oi)
3) Repeat each operation o_i, r_i times
4) Add edges between successive occurrences of each operation
5) For each edge (o_i,o_j)
   - If $H(o_i) > H(o_j)$ (slow to fast dependency)

     Add edges $(o_i^s, o_j^u), s \in \{1, 2, \dots, r_i\}, u = \left\lfloor s \times \frac{H(o_i)}{H(o_j)} \right\rfloor$
   - If $H(o_i) < H(o_j)$ (fast to slow dependency)

     Add edges $(o_i^s, o_j^u), u \in \{1, 2, \dots, r_j\},\ s = \left\lfloor u \times \frac{H(o_j)}{H(o_i)} \right\rfloor$
   - If $H(o_i) = H(o_j)$

     Add edges (o_i^s, o_j^u) between corresponding occurrences
6) For each FMU
   - Add edges between the occurrence s of the state operation and all the input and output operations of the next occurrence s+1
7) Stop when all operations and edges have been visited

MUO-RCOSIM
MULTI-CORE SCHEDULING

- Off-line heuristic approach: Similar to SynDEx (INRIA) [Grandpierre et al., 1999]
- N operations, each one:
  - Computation time
  - Earliest and latest start and end dates → Takes into account the synchronization cost
- Objective: Minimize the makespan (multiprocessor critical path) of the graph
- Cost function: Schedule pressure is the difference between:
  - Flexibility: Freedom degree of an operation: time interval inside which o_i may be executed without increasing the makespan
  - Penalty: Critical path increase by setting an operation on a processor accounting for synchronization cost

MUO-RCOSIM
MULTI-CORE SCHEDULING (CONT'D)

Multi-core scheduling heuristic

1) For each operation o_i
   - Compute S_i (resp. E_i) the earliest start (resp. end) time, and S'_i (resp. E'_i) the latest start (resp. end) time
   - Compute the flexibility $F_i = CP - E_i - E'_i$
2) Set Ω the set of operations without predecessors
3) Repeat
   - For each pair (operation o_i in Ω, core p_j)

     Compute the increase (cost) of scheduling o_i on p_j

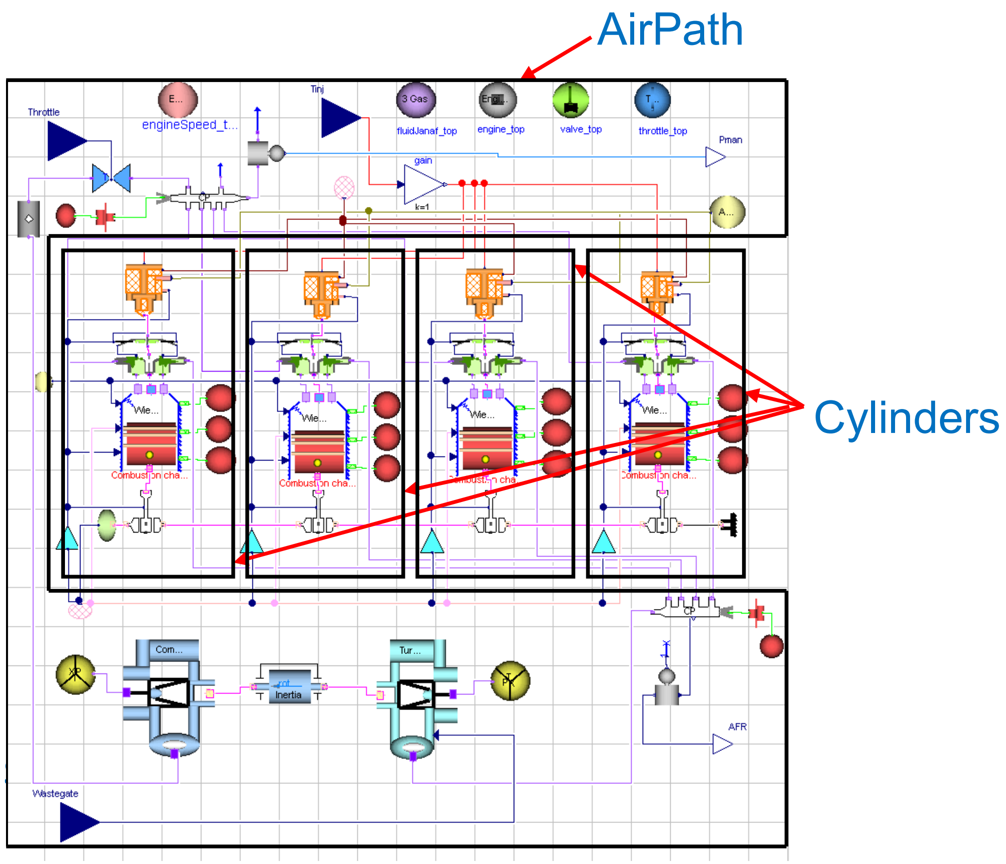
     Select for o_i, the core p_j which minimizes the cost of scheduling o_i
   - Find the operation o_i with the maximal cost on its selected core
   - Allocate o_i to its selected core
   - Remove o_i from Ω
   - Add to Ω every operation whose predecessors have been scheduled
   - Stop when all the operations have been scheduled

INSTITUT CARNOT IFPEN Transports Energie

IFP Energies nouvelles

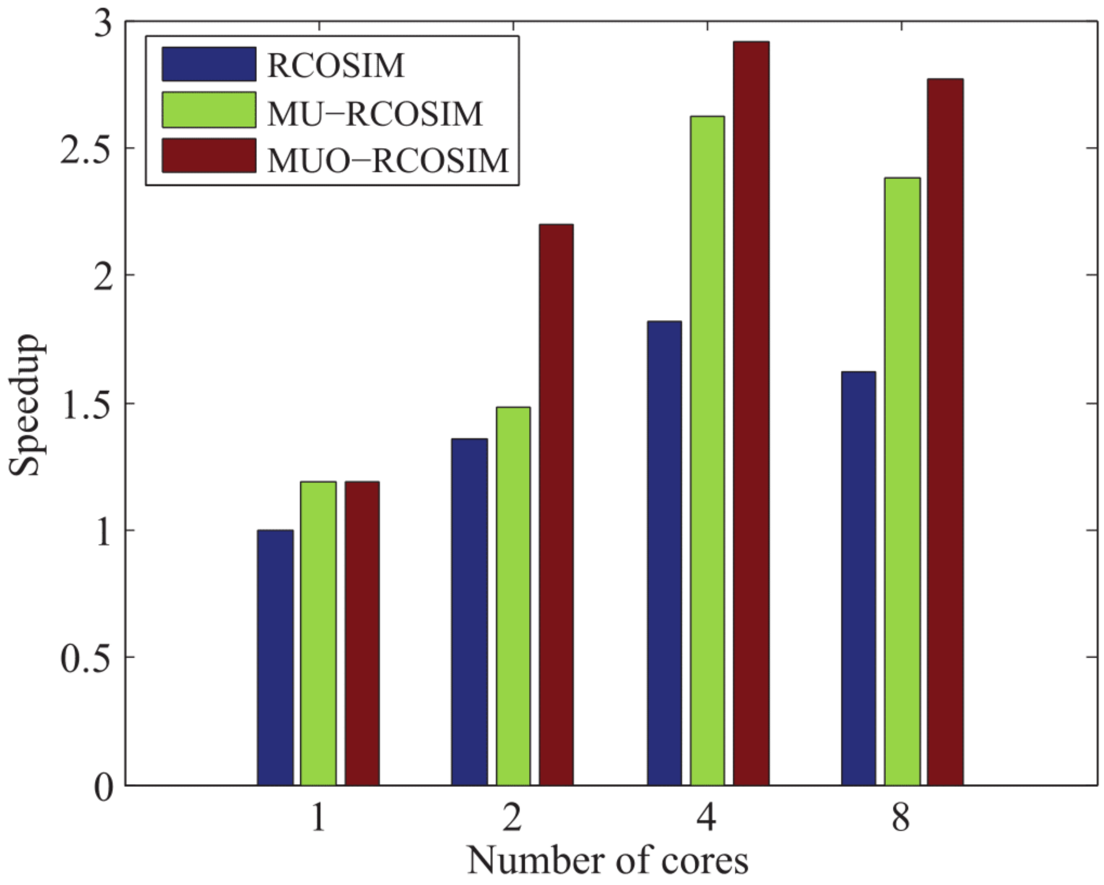
TESTS

- Case study
  - Spark Ignition RENAULT F4RT engine
  - 6 FMUs, more than 100 operations
  - Around 300 operations after applying the multi-rate transformation algorithm
    - Communication steps
      - Airpath/control: 100 µs
      - Cylinders: 20 µs
    - Integration step = communication step for all FMUs
- 3 approaches are compared
  - RCosim: Mono-rate, restricted allocations of the operations
  - MU-RCosim: **MU**lti-rate, restricted allocation of the operations
  - MUO-RCosim: **MU**lti-rate, uses the acyclic **O**rientation heuristic to handle mutual exclusion constraints



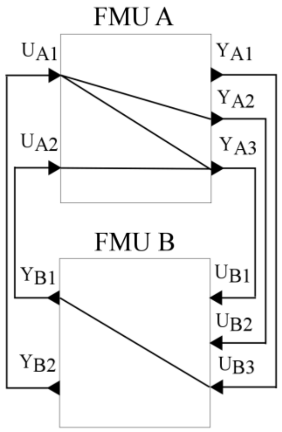
TESTS
SPEED-UP

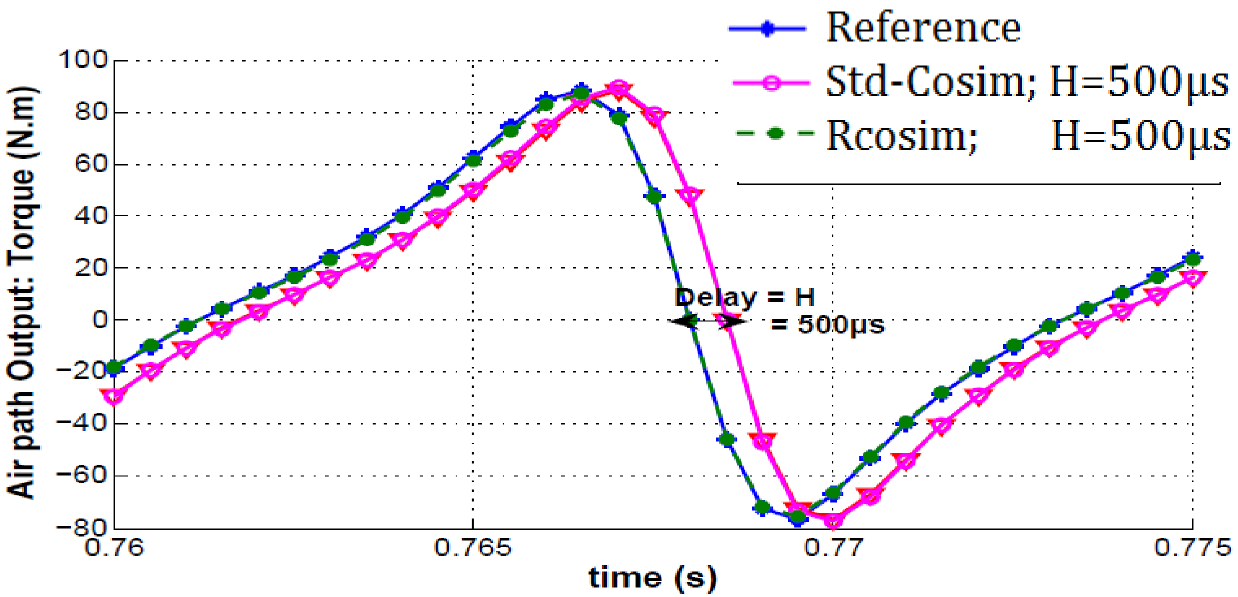
- Speed-up = $\frac{\textit{Sequential execution time}}{\textit{Parallel execution time}}$
  - Best speed-up close to 2.9 reached with 5 cores (compared to mono-core schedule)
  - MUO-RCosim > MU-RCosim > RCosim
  - Thanks to the mutual exclusion heuristic, an efficient execution order for mutual exclusive operations is defined
  - This order tends to allow the multi-core scheduling heuristic to better adapt the potential parallelism to the execution platform



RCOSIM APPROACH
ACCURACY: ELIMINATION OF DELAYS

- Torque is a direct feedthrough output: e.g. Y_{A3}
- Expected delays with Standard Co-simulation (Std-Cosim) due to arbitrary order execution decision between models
- No delays with RCosim
  - The execution order is compliant with initial model





Test with variable step solver: LSODAR

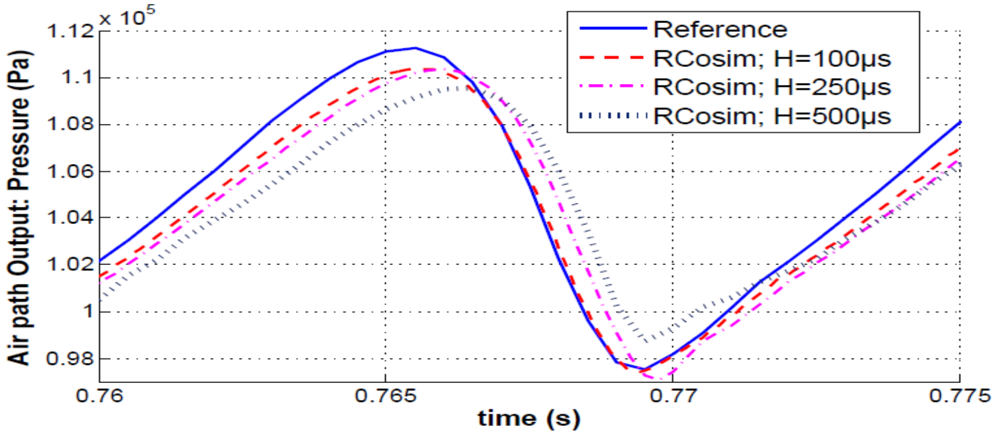
Simulation method	Std-Cosim	RCosim
Er(%) with H=100µs	2.95	0.68
Er(%) with H=250µs	9.12	1.1
Er(%) with H=500µs	**19.83**	**1.37**

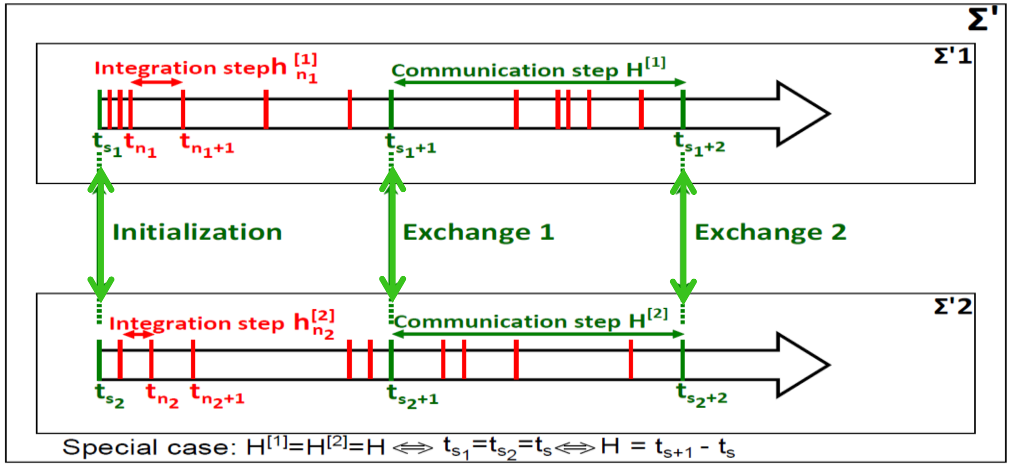
OUTLINE

- Background
  - Co-simulation: context & challenges
  - Real-time Co-Simulation: an open problem
- Improving parallelism with the RCosim approach: Refined Co-simulation
- **Ensuring co-simulation accuracy with CHOPtrey extrapolation approach**
- Mapping real-time constraints for HiL
- Future work

CONTEXT-BASED EXTRAPOLATION
IMPROVE AGAIN THE SIMULATION ACCURACY

- Limitation: with RCosim, errors are reduced but still exist
- Reason: Input data is held constant during the communication step
- Dilemma:↗↗ communication step
  - ↗↗ Integration error
  - ↗↗ Speed-up
- Idea: Extrapolate input signals to
  - Enlarge intervals
  - Reduce simulation errors





RELATED WORK

- Difficulties
  - Related work on extrapolations treated the continuous case
    - Successful for non-stiff systems / Encountered problems with stiff systems
  - Complex systems with hybrid behavior is even more difficult to predict
    - Nonlinearities, discontinuities,…
  - ➔Hard to predict the future behavior (from past observations)
    - Polynomial prediction fails due to the discontinuities
    - No universal prediction scheme, efficient with every signal
- Challenges: fast, causal and reliable prediction
  - Predictor computing cost << extra model computations with small communication steps
  - Accurate predictions for any signal (blocky/smooth; slow/steep onsets)
- Idea: Borrow the context-based approach from lossless image encoders

CHOPRED: COMPUTATIONALLY HASTY ONLINE PREDICTION
CHOPOLY: CAUSAL HOPPING OBLIVIOUS POLYNOMIALS

- $P_{\delta,\lambda,\omega}$: least squares polynomial predictor
  - δ: prediction degree;
  - λ: prediction frame length
  - ω: weighting factor
- u: input signal; τ: relative time for prediction
- Weighted moment: $\overline{m}_{d,\lambda,\omega} = \sum_{l=0}^{\lambda-1} (\lambda - l)^{\omega} l^{d} u_{-l}$
- Weighted sum of integer powers: $\overline{z}_{d,\lambda,\omega} = \sum_{l=0}^{\lambda-1} (\lambda - l)^{\omega} l^{d}$
- General formula for extrapolation:
  - Use of LUT ➔ fast computation

$$u(\tau) = \begin{bmatrix} 1 & \tau & \cdots & \tau^{\delta} \end{bmatrix} \begin{bmatrix} \overline{z}_{0,\lambda,\omega} & -\overline{z}_{1,\lambda,\omega} & \cdots & (-1)^{\delta}\overline{z}_{\delta,\lambda,\omega} \\ -\overline{z}_{1,\lambda,\omega} & & & \vdots \\ \vdots & & & \vdots \\ (-1)^{\delta}\overline{z}_{\delta,\lambda,\omega} & \cdots & \cdots & \overline{z}_{2\delta,\lambda,\omega} \end{bmatrix}^{-1} \begin{bmatrix} \overline{m}_{0,\lambda,\omega} \\ -\overline{m}_{1,\lambda,\omega} \\ \vdots \\ (-1)^{\delta}\overline{m}_{\delta,\lambda,\omega} \end{bmatrix}$$

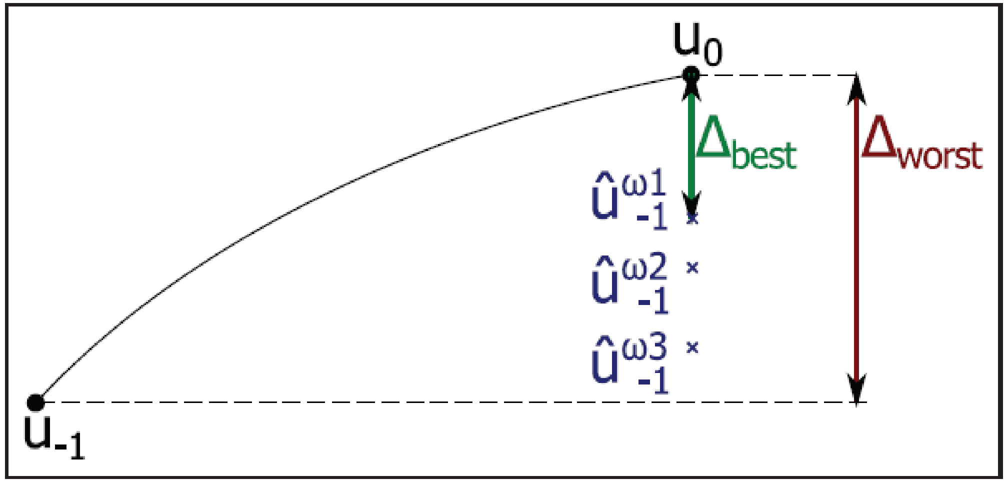
INSTITUT CARNOT IFPEN Transports Energie

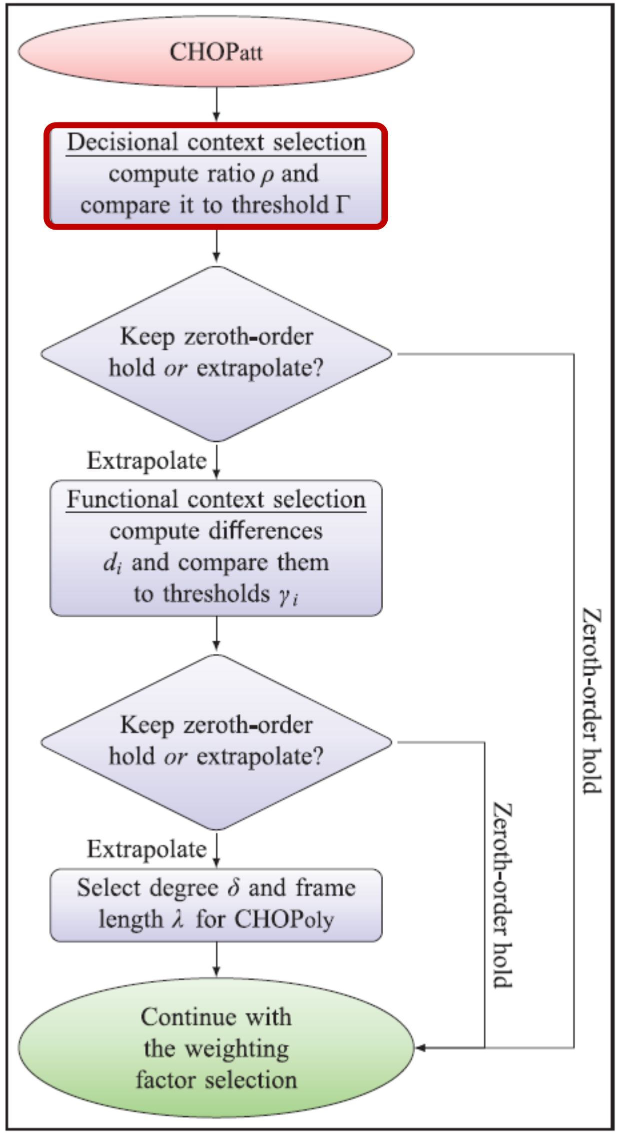
IFP Energies nouvelles

CHOPATT: CONTEXTUAL AND HIERARCHICAL ONTOLOGY OF PATTERNS

META- OR DECISIONAL CONTEXT SELECTION

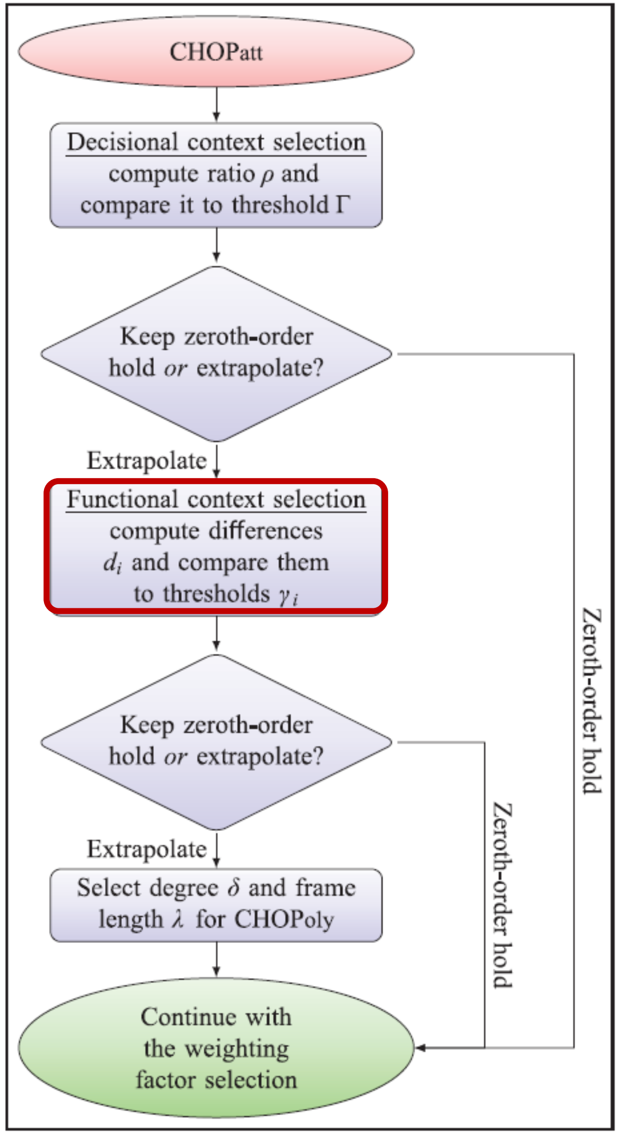
- Worst case scenario without extrapolation: $\Delta_{\text{worst}} = |u_0 - u_{-1}|$
- Best prediction pattern: $\Delta_{\text{best}} = \min_{\omega \in \Omega} |u_0 - \hat{u}^{\omega}_{-1}|$; $\Omega = \{0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$
- Ratio: $\rho = \frac{\Delta_{\text{best}}}{\Delta_{\text{worst}}}$
- Threshold: $0.7 \leqslant \Gamma < 1$ e.g. $\Gamma = 90\%$
- If $\rho > \Gamma$ then sharp and fast variation → Select the decisional context: cliff context





CHOPATT: CONTEXTUAL AND HIERARCHICAL ONTOLOGY OF PATTERNS
FUNCTIONAL CONTEXT SELECTION

- Differences (variations): $d_0 = u_0 - u_{-1}$ and $d_{-1} = u_{-1} - u_{-2}$
- Thresholds: $\gamma_0 = \gamma_{-1} = \frac{1}{2} \max_{i \in [1-\Lambda, \dots, -3]} (|u_i - u_{i+1}|)$
- Conditions: O if $|d_i| = 0$; C_i if $0 < |d_i| \leqslant \gamma_i$; $\overline{C_i}$ if $|d_i| > \gamma_i$.

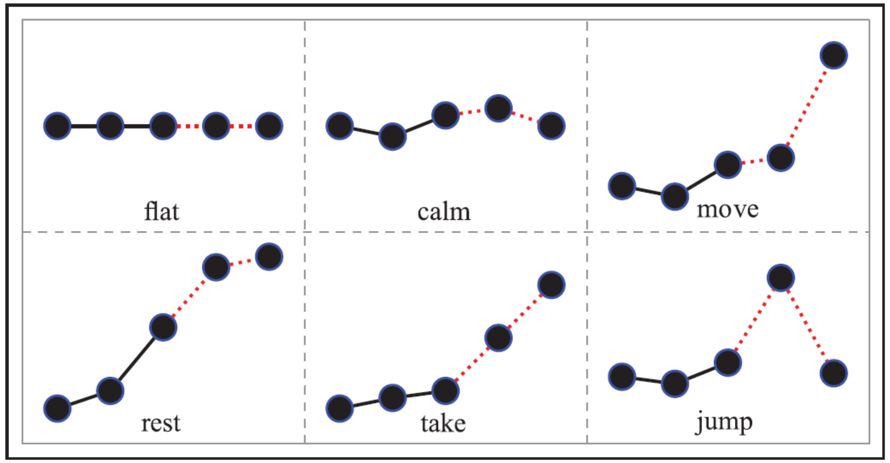


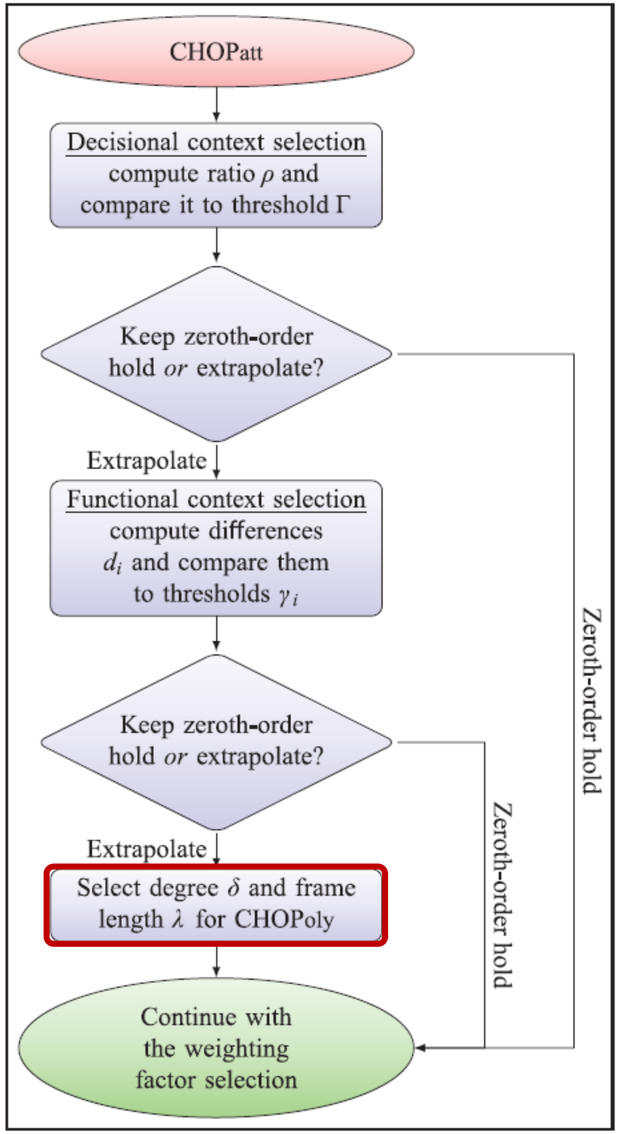
INSTITUT CARNOT IFPEN Transports Energie
IFP Energies nouvelles

CHOPATT: CONTEXTUAL AND HIERARCHICAL ONTOLOGY OF PATTERNS
FUNCTIONAL CONTEXT SELECTION

- Differences (variations): $d_0 = u_0 - u_{-1}$ and $d_{-1} = u_{-1} - u_{-2}$
- Thresholds: $\gamma_0 = \gamma_{-1} = \frac{1}{2} \max_{i \in [1-\Lambda,\ldots,-3]} (|u_i - u_{i+1}|)$
- Conditions: O if $|d_i| = 0$; C_i if $0 < |d_i| \leqslant \gamma_i$; $\overline{C_i}$ if $|d_i| > \gamma_i$.

n(ame)	#	$\|d_{-1}\|$	$\|d_0\|$	$d_{-1}.d_0$	(δ, λ, ω)
f(lat)	0	O	O	O	(0, 1, .)
c(alm)	1	C_1	C_2	any	(2, 5, .)
m(ove)	2	C_1	$\bar{C}_2$	any	(0, 1, .)
r(est)	3	$\bar{C}_1$	C_2	any	(0, 2, .)
t(ake)	4	$\bar{C}_1$	$\bar{C}_2$	> 0	(1, 3, .)
j(ump)	5	$\bar{C}_1$	$\bar{C}_2$	< 0	(0, 1, .)





INSTITUT CARNOT IFPEN Transports Energie
IFP Energies nouvelles

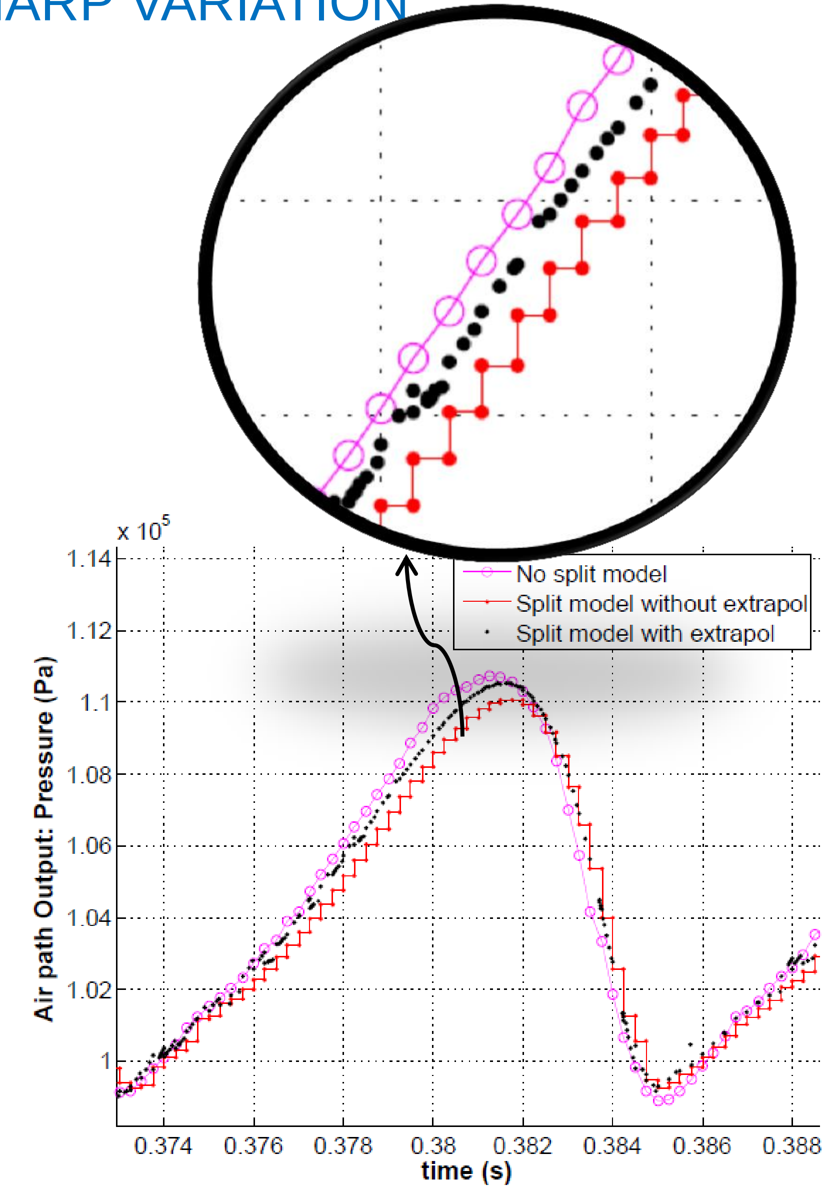
SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC DETECTION OF SHARP VARIATION

- Same case study
  - 118 states/312 events
  - Solver: LSODAR
  - Communication step: 200µs
- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC DETECTION OF SHARP VARIATION

- Same case study
  - 118 states/312 events
  - Solver: LSODAR
  - Communication step: 200µs
- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

➔ Context-based extrapolation?



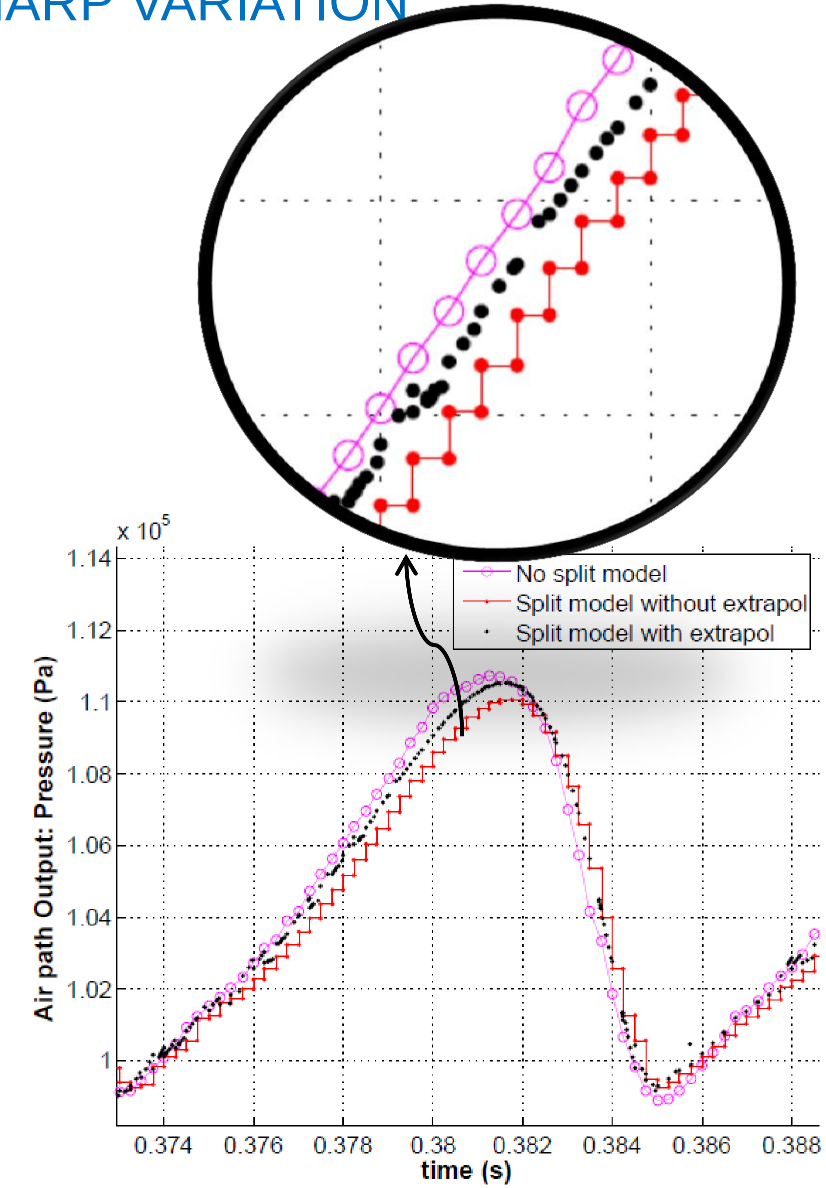
INSTITUT CARNOT
IFPEN Transports Energie

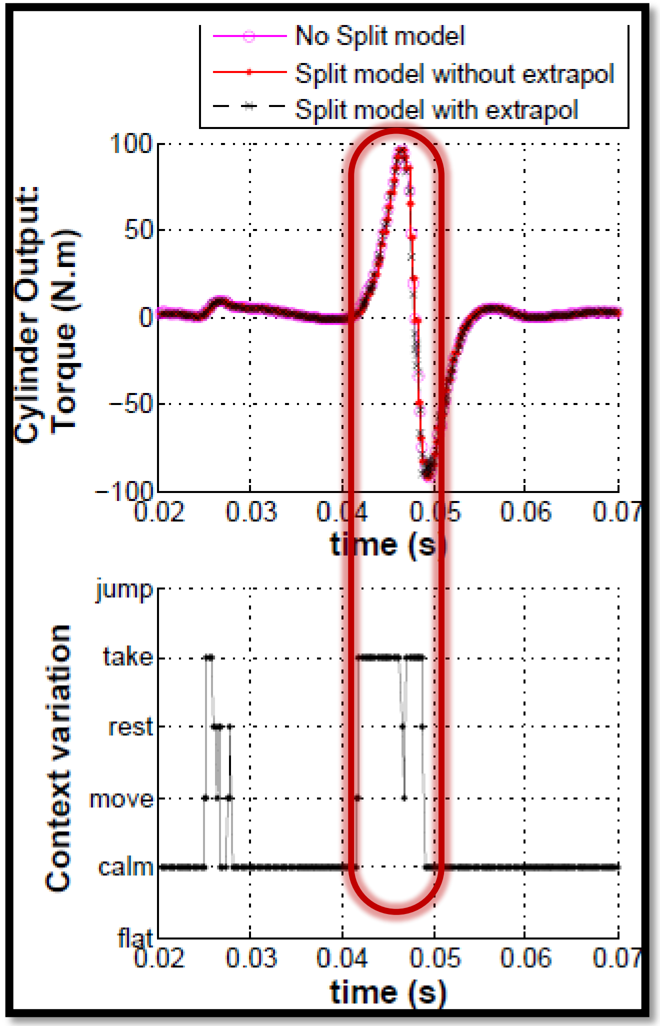
Energies nouvelles

SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC DETECTION OF SHARP VARIATION

- Same case study
  - 118 states/312 events
  - Solver: LSODAR
  - Communication step: 200µs
- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

→ Context-based extrapolation?





INSTITUT CARNOT IFPEN Transports Energie
IFP Energies nouvelles

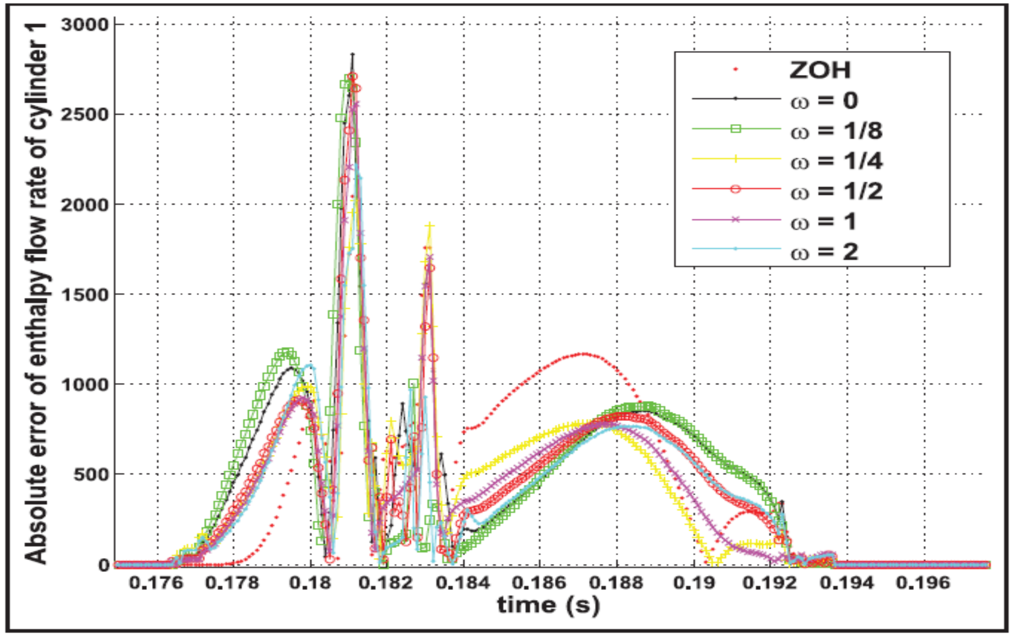
SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC SELECTION OF THE WEIGHTING FACTOR

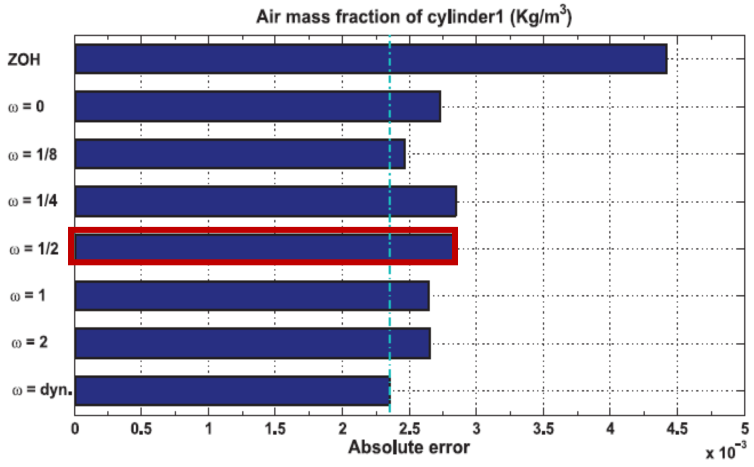
- No unique best weighting factor ω due to complex coupled systems

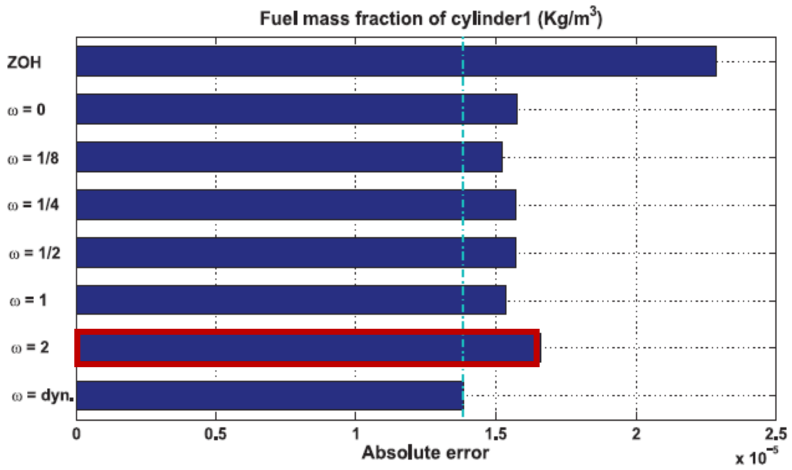
➔ Dynamic selection of ω

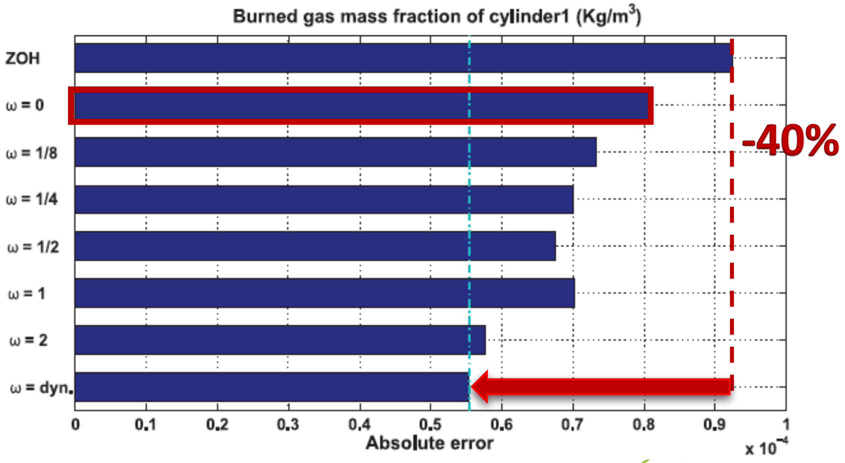
- At each communication step, ω_{best} is selected and used for the current step

→ Cumulative integration error is the lowest one









CHOPtrey PERFORMANCE
SPEED-UP VERSUS ACCURACY

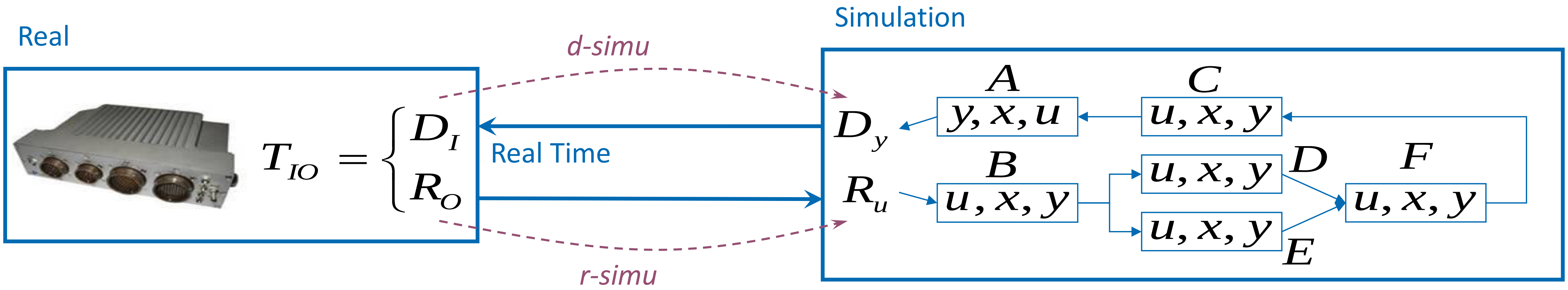
- The speed-up factor is compared with single-threaded reference
- The model is split into 5 threads integrated in parallel on 5 cores
  - Containment of events detection handling → solvers accelerations → overcompensate multi-threading costs
- The relative error variation is compared with ZOH at 100 µs

Communication step	Prediction	Speed-up factor	Relative error variation (%)	
			Burned gas density	Fuel density
100 µs	ZOH	8.9	-	-
250 µs	ZOH	10.01 (+12.5%)	7	341
	CHOPtrey	10.07	-26	21

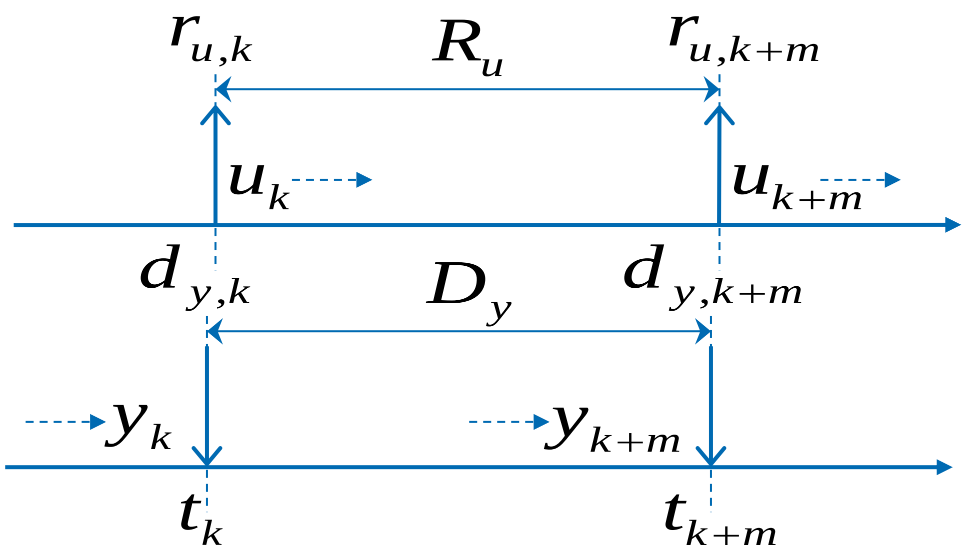
OUTLINE

- Background
  - Co-simulation: context & challenges
  - Real-time Co-Simulation: an open problem
- Improving parallelism with the RCosim approach: Refined Co-simulation
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- **Mapping real-time constraints for HiL**
- Future work

REAL TIME SIMULATION
FROM REAL TIME TO SIMULATED TIME



- Relative simulation constraints
- At each interaction : $n \times T_{IO} = t_k$
  - y_k required → deadline $d_{y,k}$
  - u_k available→ release $r_{u,k}$
- Impact on the underlying computations
  - Linked to the nature of the models and their interconnection
  - How to propagate when step time (h) are different for models A, B C, …?



INSTITUT CARNOT IFPEN Transports Energie

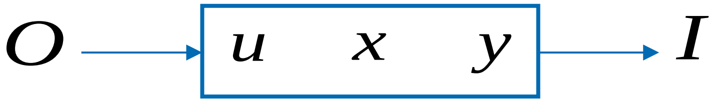
INTRA-MODEL PROPAGATION AND « DIRECT FEEDTHROUGH »

Absolute Simu constraints with offset

$$R_v, D_v = (\Phi, T)$$

offset period

$$n \times T_{IO} = t_k$$



$$DF \Leftrightarrow u_k \rightarrow y_k$$

- Propagation *r-simu* $R_u = R_O$
  - R_x : offset h
  - $DF \Rightarrow R_y = R_u$
  -

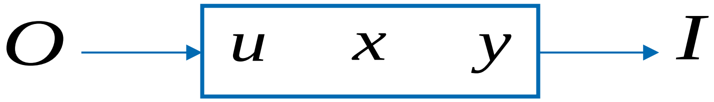
INTRA-MODEL PROPAGATION AND « DIRECT FEEDTHROUGH »

Absolute Simu constraints with offset

$$R_v, D_v = (\Phi, T)$$

offset — period

$$n \times T_{IO} = t_k$$



$$NDF \Leftrightarrow u_k \nrightarrow y_k$$

- Propagation *r-simu* $R_u = R_O$
  - R_x : offset h
  - $DF \Rightarrow R_y = R_u$
  - $NDF \Rightarrow R_y$: offset h

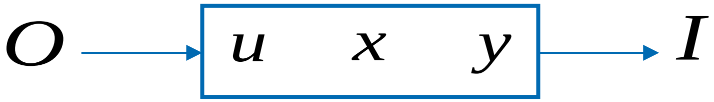
INTRA-MODEL PROPAGATION AND « DIRECT FEEDTHROUGH »

Absolute Simu constraints with offset

$$R_v, D_v = (\Phi, T)$$

offset (Φ) — period (T)

$$n \times T_{IO} = t_k$$



$$DF \Leftrightarrow u_k \rightarrow y_k$$

- Propagation *r-simu* $R_u = R_O$
  - R_x : offset h
  - $DF \Rightarrow R_y = R_u$
  - $NDF \Rightarrow R_y$: offset h
- Propagation *d-simu* $D_y = D_I$
  - D_x : no offset
  - $DF \Rightarrow D_u = D_y$
  -

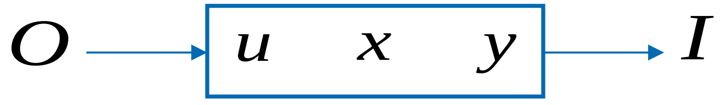
INTRA-MODEL PROPAGATION AND « DIRECT FEEDTHROUGH »

Absolute Simu constraints with offset

$$R_v, D_v = (\Phi, T)$$

offset period

$$n \times T_{IO} = t_k$$



$$NDF \Leftrightarrow u_k \not\to y_k$$

- Propagation *r-simu* $R_u = R_O$
  - R_x : offset h
  - $DF \Rightarrow R_y = R_u$
  - $NDF \Rightarrow R_y$: offset h

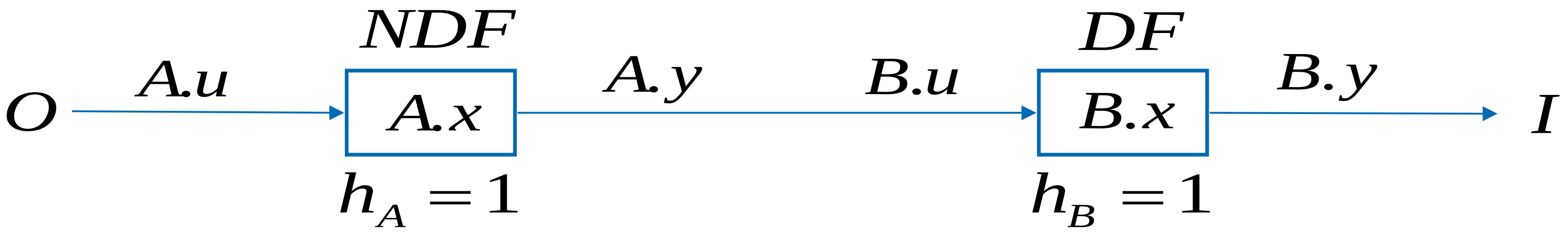
- Propagation *d-simu* $D_y = D_I$
  - D_x : no offset
  - $DF \Rightarrow D_u = D_y$
  - $NDF \Rightarrow D_u$: offset $-h$

INSTITUT CARNOT IFPEN Transports Energie

Energies nouvelles

EXAMPLE OF PROPAGATION
RELATIVE MESHES

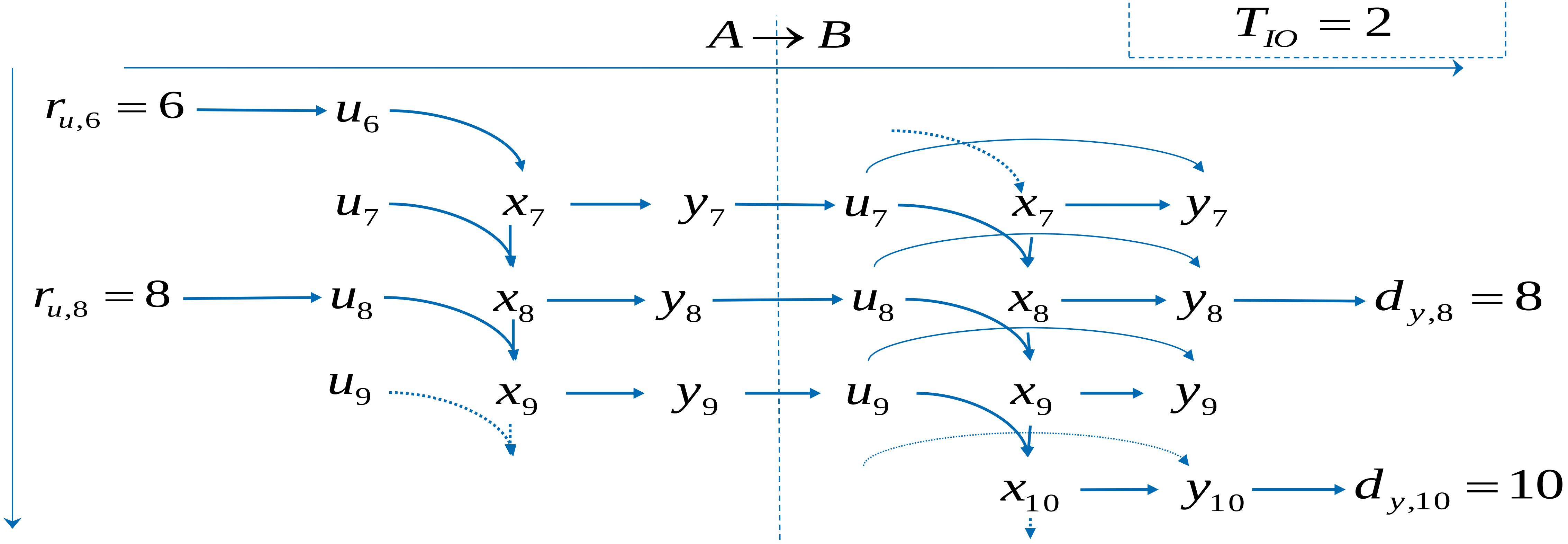
$T_{IO} = 2$



(Φ, T) \ v	$A.u$	$A.x$	$A.y = B.u$	$B.x$	$B.y$
R_v	$R_O = (0,2)$	$(1,2)$	$(1,2)$	$(2,2)$	$(1,2)$
D_v	$(-1,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$D_I = (0,2)$

EXAMPLE OF PROPAGATION
ABSOLUTE SIMULATION CONSTRAINTS

$$h_A = h_B = 1$$
$$T_{IO} = 2$$

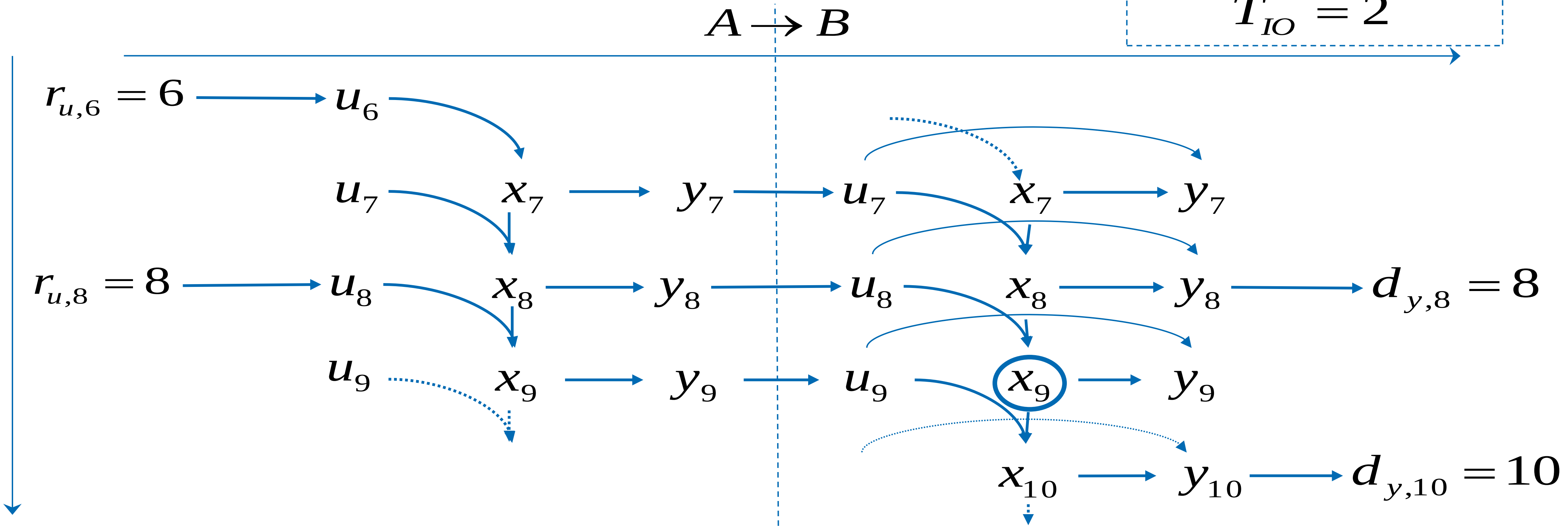


$$\forall D_v = (\Phi, T),\ d_{v,i} = \left\lceil \frac{h \times i - \Phi}{T} \right\rceil \times T$$

$$\forall R_v = (\Phi, T),\ r_{v,i} = \left\lfloor \frac{h \times i - \Phi}{T} \right\rfloor \times T$$

EXAMPLE OF PROPAGATION
ABSOLUTE SIMULATION CONSTRAINTS

$$h_A = h_B = 1$$
$$T_{IO} = 2$$



$$\forall D_v = (\Phi, T),\ d_{v,i} = \left\lceil \frac{h \times i - \Phi}{T} \right\rceil \times T$$

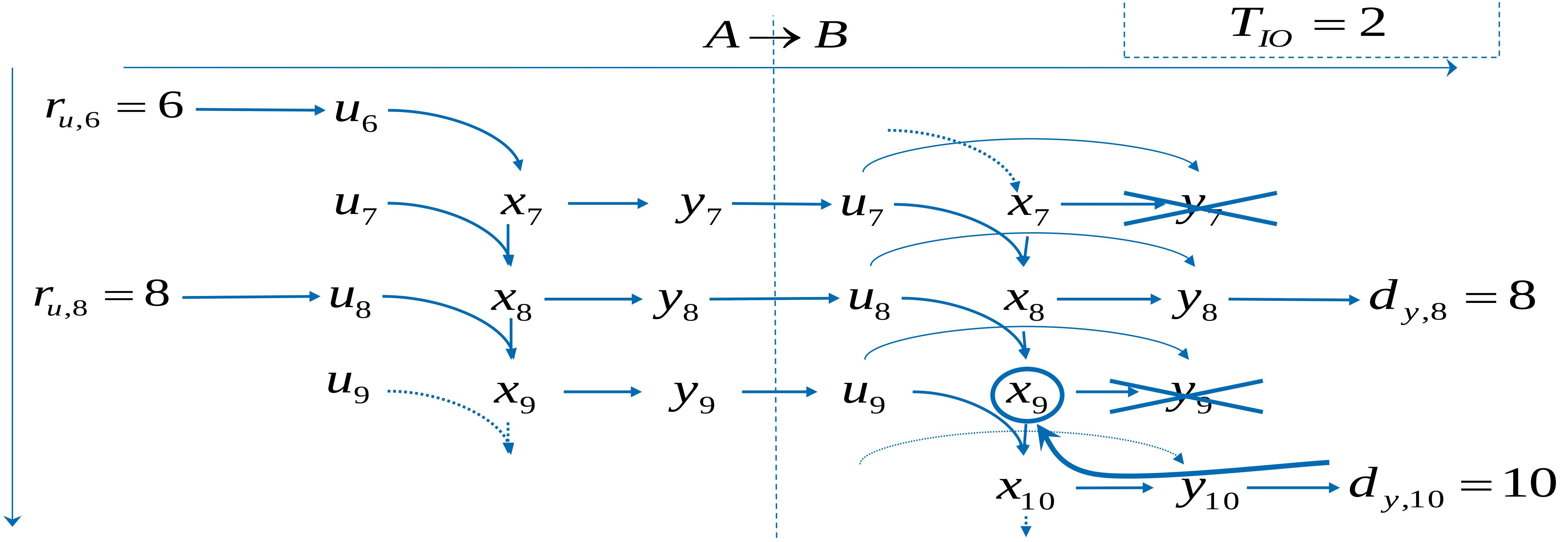
$$\forall R_v = (\Phi, T),\ r_{v,i} = \left\lfloor \frac{h \times i - \Phi}{T} \right\rfloor \times T$$

	$B.x$
R_v	(2,2)
D_v	(0,2)

$k = 9$

EXAMPLE OF PROPAGATION
ABSOLUTE SIMULATION CONSTRAINTS

$$h_A = h_B = 1$$
$$T_{IO} = 2$$



$$\forall D_v = (\Phi, T),\ d_{v,i} = \left\lceil \frac{h \times i - \Phi}{T} \right\rceil \times T$$

$$\forall R_v = (\Phi, T),\ r_{v,i} = \left\lfloor \frac{h \times i - \Phi}{T} \right\rfloor \times T$$

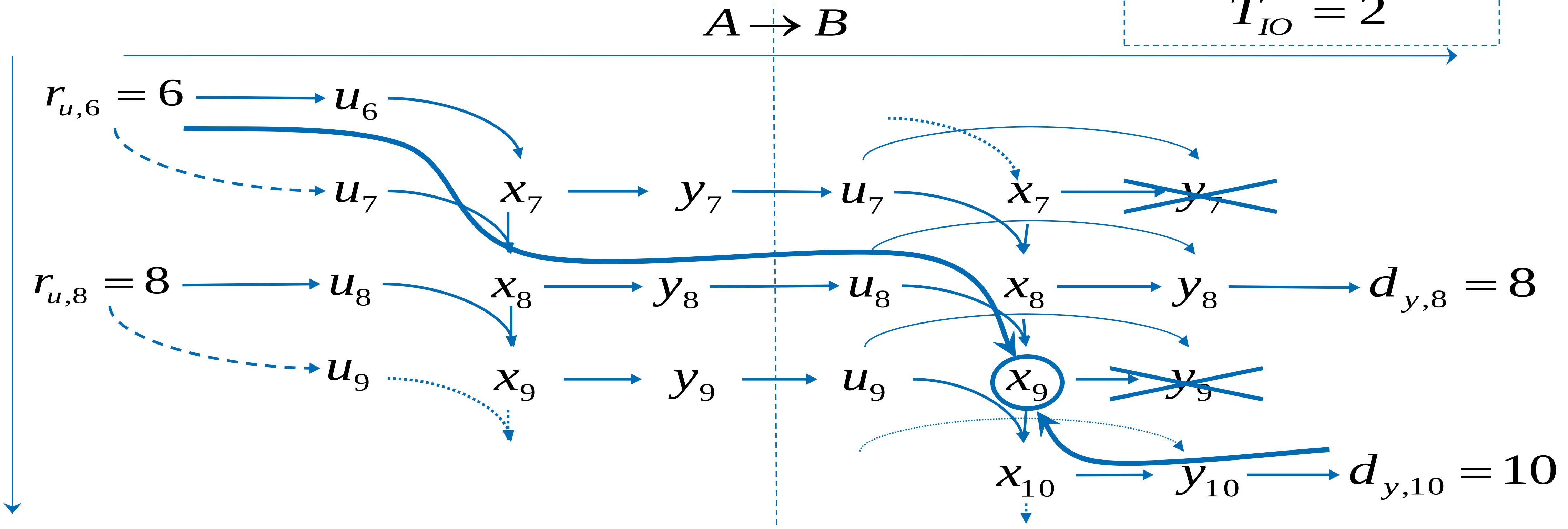
	$B.x$
R_v	$(2,2)$
D_v	$(0,2)$

$$k = 9$$

$$d_{B.x,9} = \left\lceil \frac{1 \times 9 - 0}{2} \right\rceil \times 2 = 10$$

EXAMPLE OF PROPAGATION
ABSOLUTE SIMULATION CONSTRAINTS

$$h_A = h_B = 1$$
$$T_{IO} = 2$$



$$\forall D_v = (\Phi, T),\ d_{v,i} = \left\lceil \frac{h \times i - \Phi}{T} \right\rceil \times T$$

$$\forall R_v = (\Phi, T),\ r_{v,i} = \left\lfloor \frac{h \times i - \Phi}{T} \right\rfloor \times T$$

	$B.x$
R_v	(2,2)
D_v	(0,2)

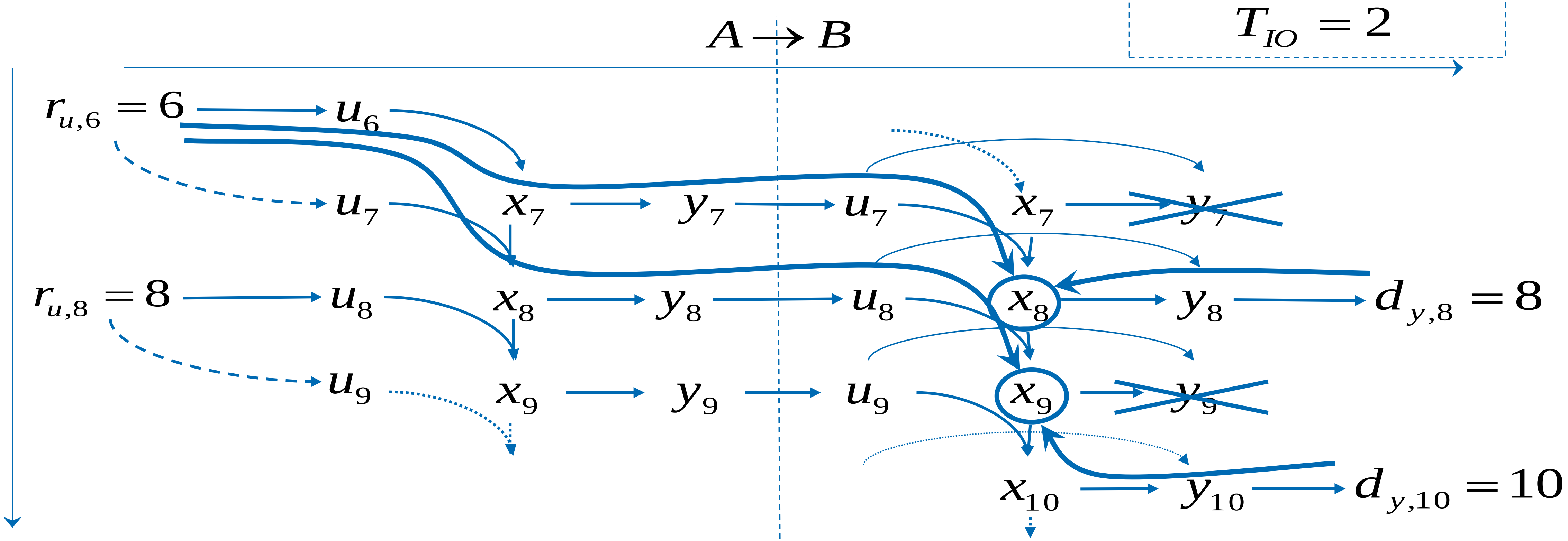
$k = 9$

$$d_{B.x,9} = \left\lceil \frac{1 \times 9 - 0}{2} \right\rceil \times 2 = 10$$

$$r_{B.x,9} = \left\lfloor \frac{1 \times 9 - 2}{2} \right\rfloor \times 2 = 6$$

EXAMPLE OF PROPAGATION
ABSOLUTE SIMULATION CONSTRAINTS

$$h_A = h_B = 1$$
$$T_{IO} = 2$$



$$d_{B.x,8} = \left\lceil \frac{1 \times 8 - 0}{2} \right\rceil \times 2 = 8$$

$$r_{B.x,8} = \left\lfloor \frac{1 \times 8 - 2}{2} \right\rfloor \times 2 = 6$$

2

$$d_{B.x,9} = \left\lceil \frac{1 \times 9 - 0}{2} \right\rceil \times 2 = 10$$

$$r_{B.x,9} = \left\lfloor \frac{1 \times 9 - 2}{2} \right\rfloor \times 2 = 6$$

4

FUTURE WORK

- Short term
  - Comparison of RCOSIM heuristics with an exact scheduling algorithm
  - Comparison of RCOSIM offline approach with on-line scheduling
- Mid term
  - Extension of real-time constraints propagation rules to the RCOSIM fine-grained approach
  - Real-time multicore scheduling heuristics for an HiL version of the RCOSIM

INSTITUT CARNOT
IFPEN Transports Energie

REFERENCES

- S. Saidi et al., “Automatic Parallelization of Multi-Rate FMI-based Co-Simulation on Multi-Core” *Spring Simulation Multiconference* (Apr. 2017)
- A. Ben Khaled-El Feki et al., “CHOPtrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex system”, *Simulation*, 2017, vol. 93(3), pp. 185-200. DOI : 10.1177/0037549716684026
- A. Ben Khaled-El Feki et al., “Fast multi-core co-simulation of Cyber-Physical Systems: application to internal combustion engines”, *Simulation Modelling Practice and Theory*, 2014, vol. 47, pp. 79-91. DOI : 10.1016/J.SIMPAT.2014.05.002