EFFICIENT EXTRAPOLATION FOR PARALLEL CO-SIMULATION OF COUPLED SYSTEMS

(CHOPtrey)

Abir Ben Khaled - El Feki, Laurent Duval, Mongi Ben Gaid

OUTLINE

- **Background on co-simulation: context & challenges**
- Results from previous work
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- Conclusion and perspectives

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

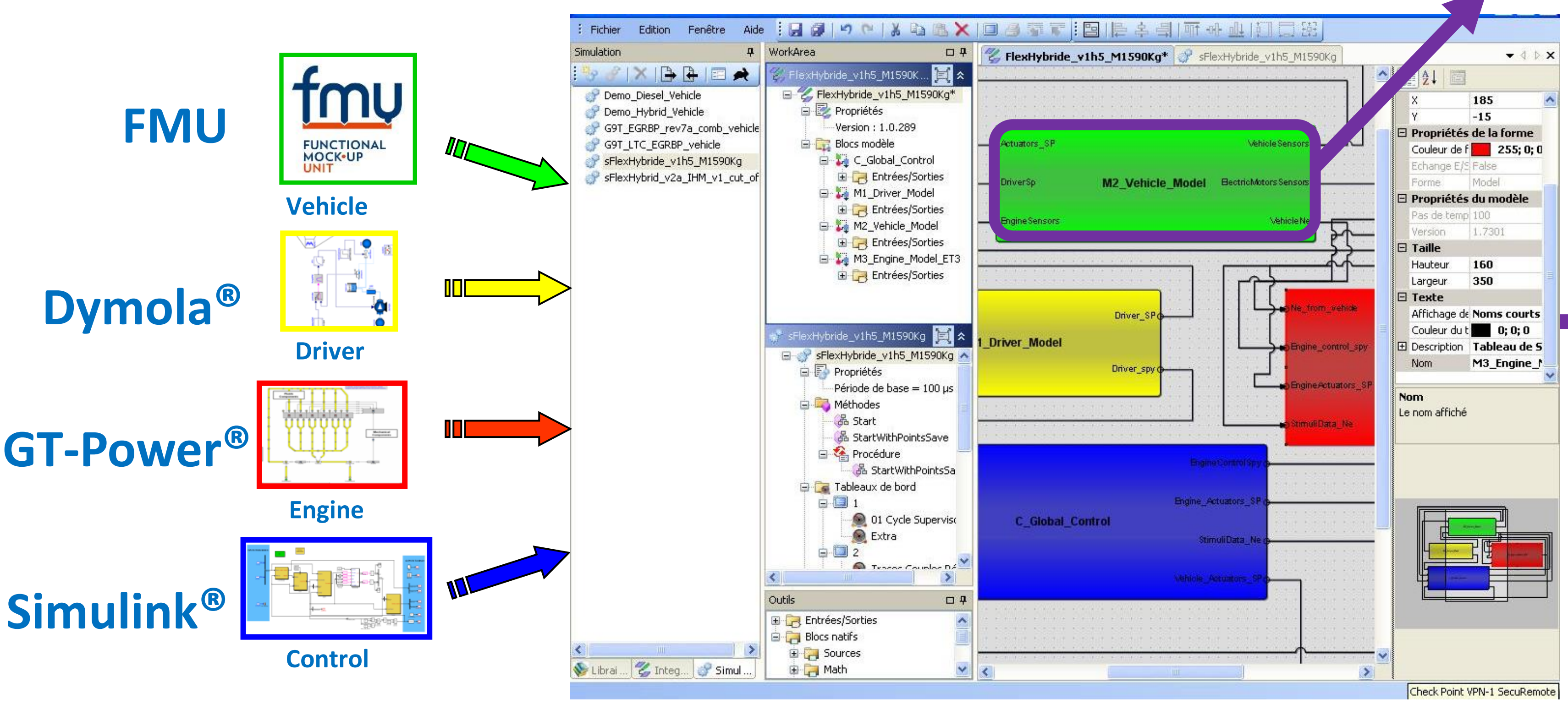
BACKGROUND (CONT'D)

- A multi-core co-simulation kernel: Why?
  - System-level simulation leads to put together models which are classically disconnected, increasing the CPU demand at simulation time
  - Simulation time becomes an important metric for model complexity
  - Most 0D/1D simulation tools have mono-core kernel and doesn't exploit available parallelism provided by multi-core computers

How long will this CPU power remain unused ?

BACKGROUND (CONT'D)

- Is integrated with its <u>own solver</u>
- Communicates its data at its <u>own rate</u>



ASAP simulation or HiL

Acceleration thanks to multi-core

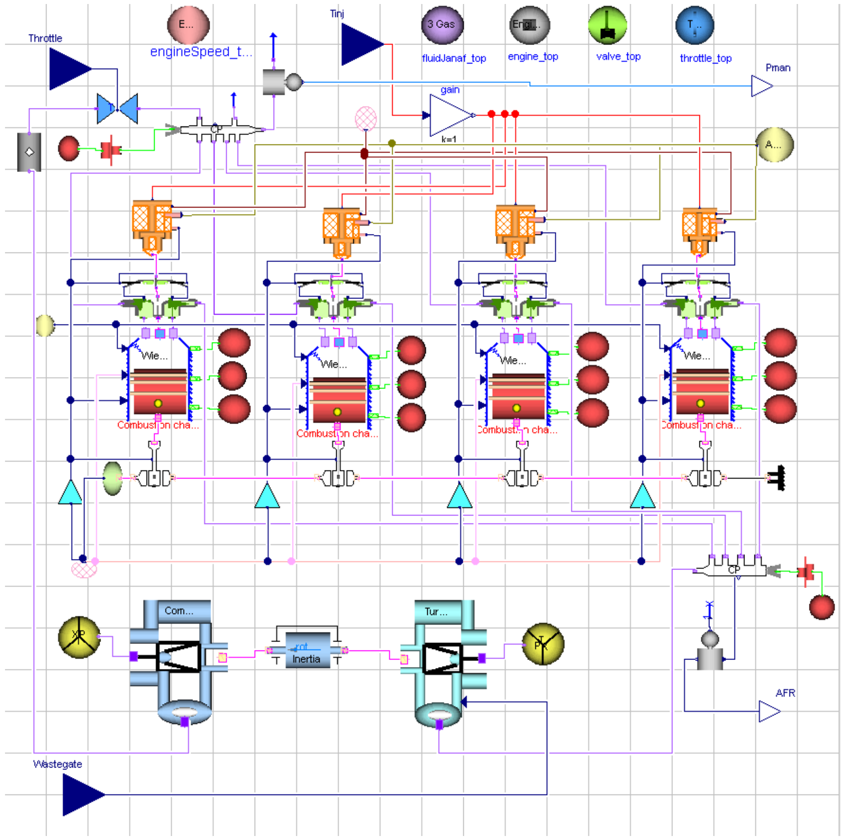
xMOD™ IFPEN co-simulation software

OUTLINE

- Background on co-simulation: context & challenges
- **Results from previous work**
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- Conclusion and perspectives

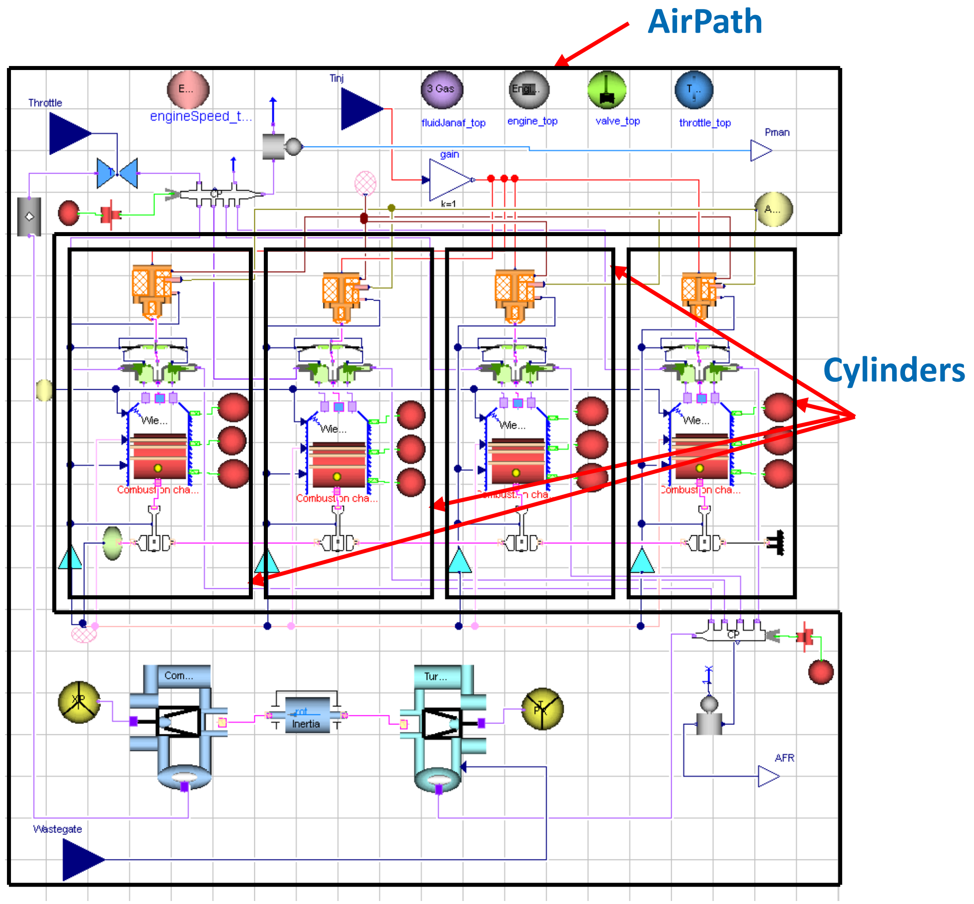
RESULTS FROM PREVIOUS WORK

- **Case study: Engine simulator**
  - Spark Ignition engine (Renault)
    - 4 cylinders + Airpath
    - 118 state variables
    - 312 event indicators
  - Modeling & simulation tools
    - Dymola (ModEngine library)
    - xMOD (FMUs)
  - Solver
    - LSODAR: Root-finding / Stiffness detection



RESULTS FROM PREVIOUS WORK

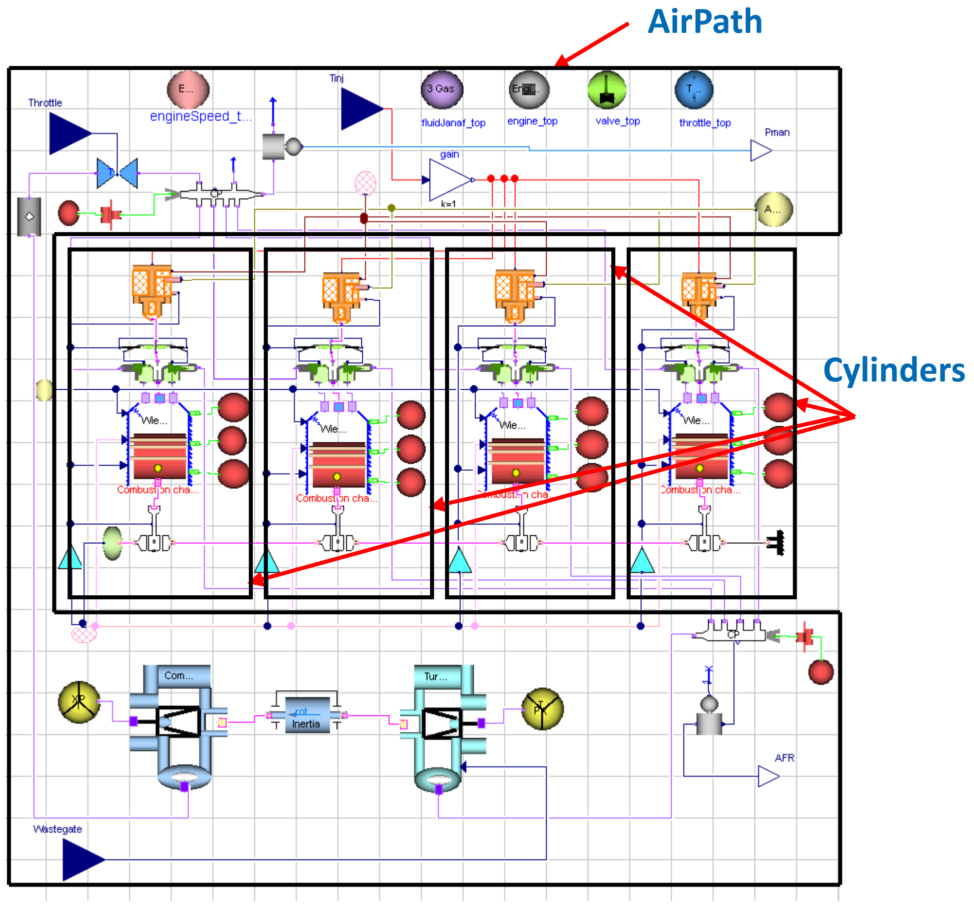
- **Case study: Engine simulator**
  - Spark Ignition engine (Renault)
    - 4 cylinders + Airpath
    - 118 state variables
    - 312 event indicators
  - Modeling & simulation tools
    - Dymola (ModEngine library)
    - xMOD (FMUs)
  - Solver
    - LSODAR: Root-finding / Stiffness detection
  - Multi-core simulation
    - 5 components on 5 cores



INSTITUT CARNOT IFPEN Transports Energie
IFP Energies nouvelles

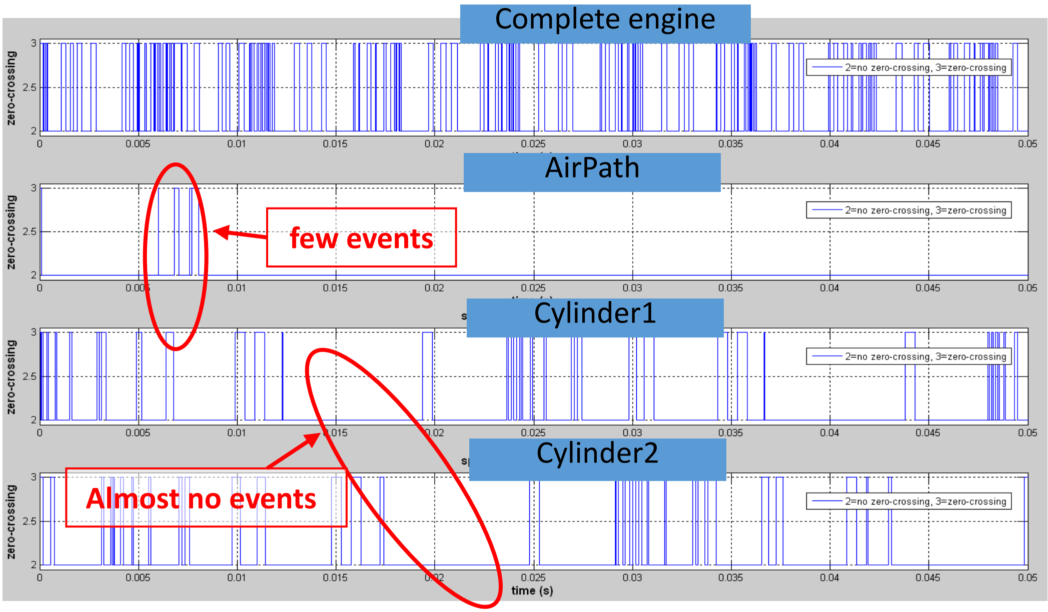
RESULTS FROM PREVIOUS WORK

- **Case study: Engine simulator**
  - Spark Ignition engine (Renault)
    - 4 cylinders + Airpath
    - 118 state variables
    - 312 event indicators
  - Modeling & simulation tools
    - Dymola (ModEngine library)
    - xMOD (FMUs)
  - Solver
    - LSODAR: Root-finding / Stiffness detection
  - Multi-core simulation
    - 5 components on 5 cores
- **Splitting is speed-up**
  - Events are related usually to the evolution of a subset of the state vector
  - Discontinuities are independent from a physical point of view



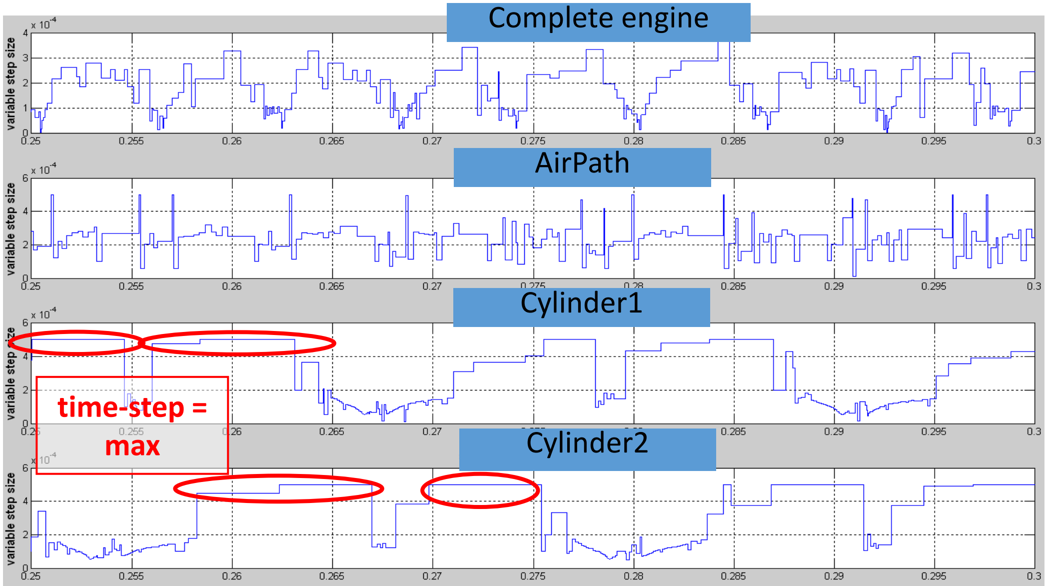
RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

- Number of events is reduced locally



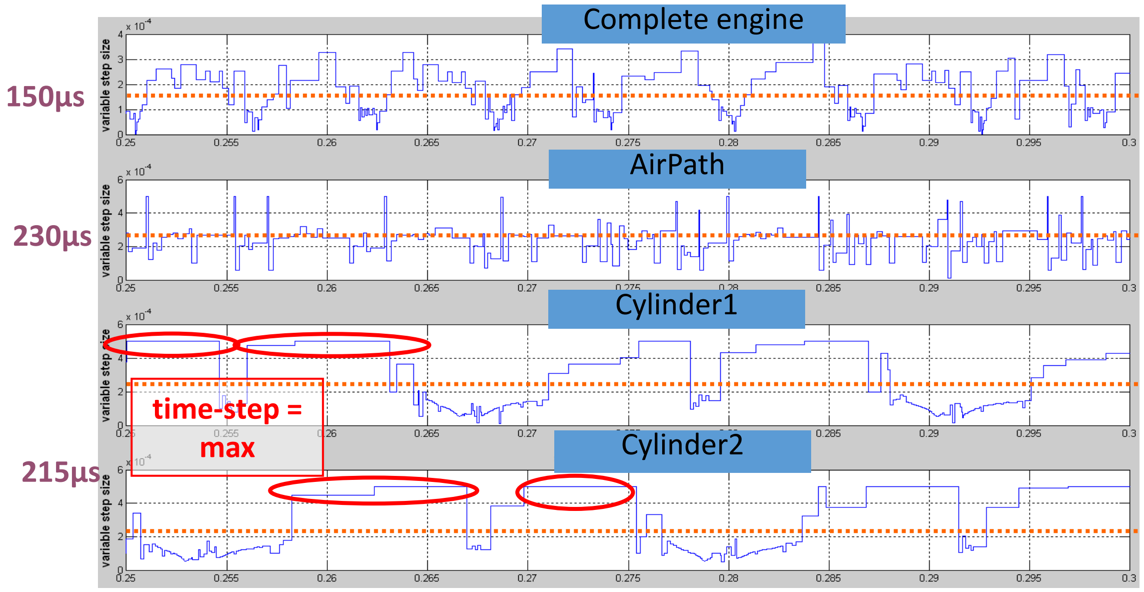
RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)



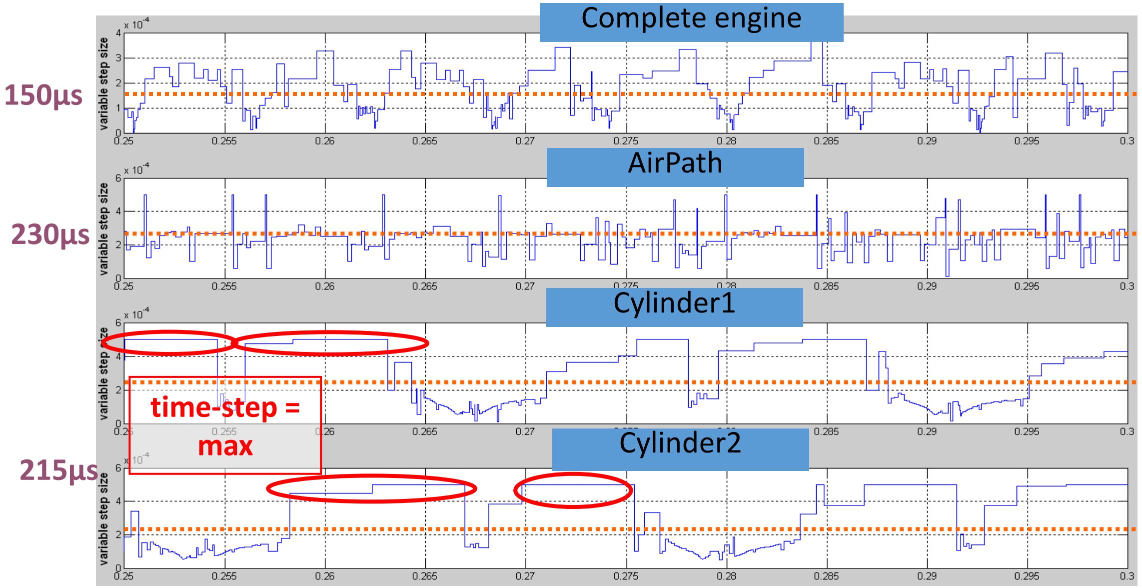
RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)
  - ➔Mean value increased from **150µs** to **230µs**



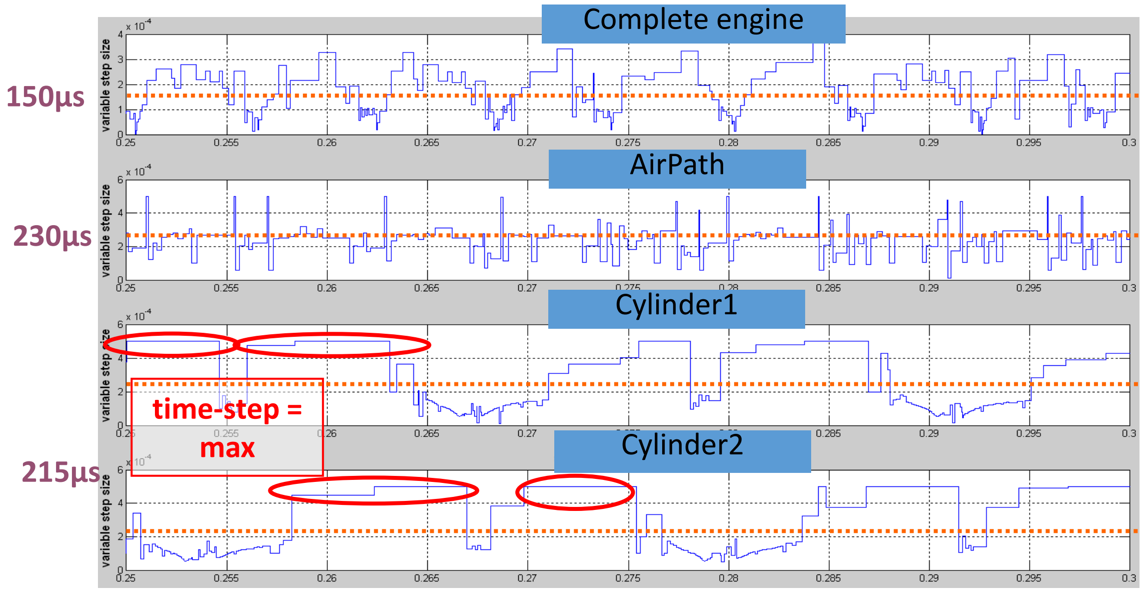
RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

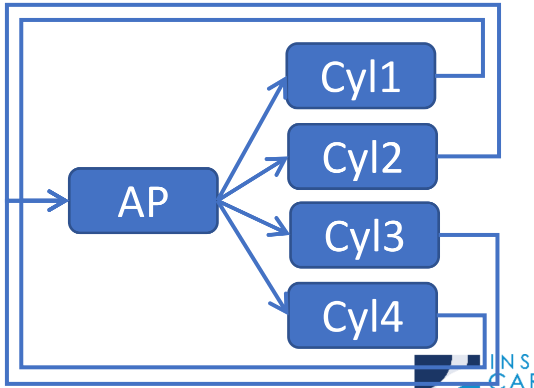
- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)
  - ➔Mean value increased from **150µs** to **230µs**
- Result on speed-up
  - Mono-core simulation
    - 5 threads on 1 core
    - Speed-up ≈ **2**
    - Thanks to System splitting & Solver coupling
    - Despite multi threading cost



RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

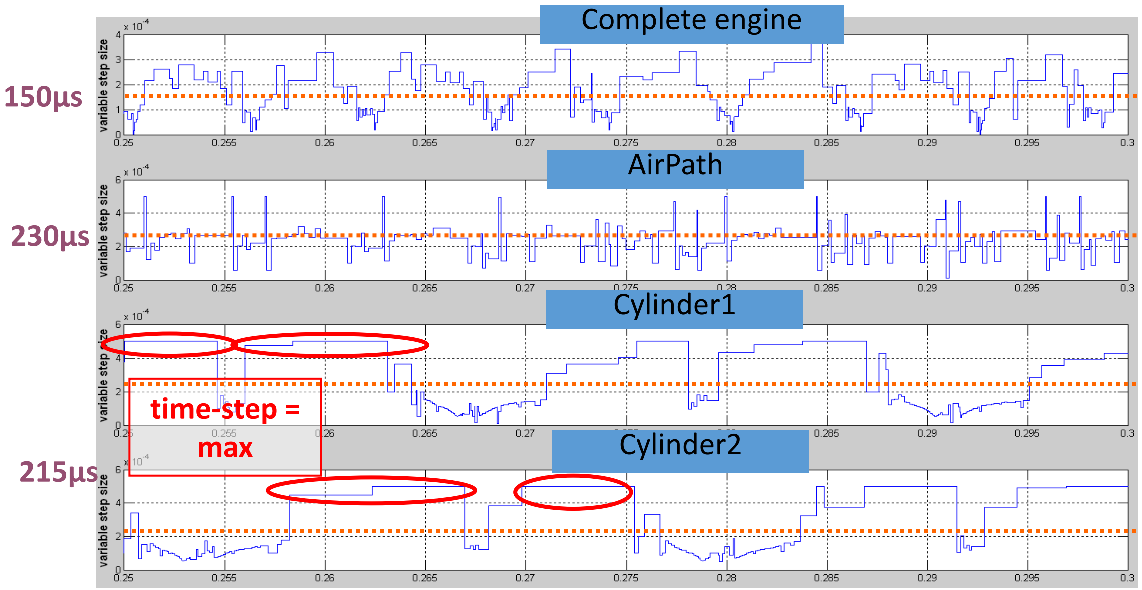
- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)
  - ➔Mean value increased from **150µs** to **230µs**
- Result on speed-up
  - Mono-core simulation
    - 5 threads on 1 core
    - Speed-up ≈ **2**
    - Thanks to System splitting & Solver coupling
    - Despite multi threading cost
  - Multi-core simulation
    - 5 threads on 5 cores

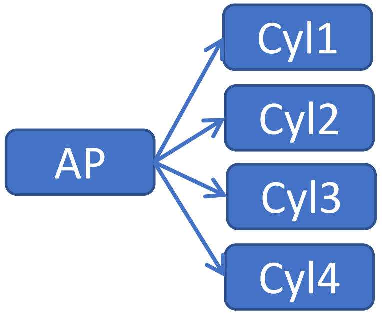




RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

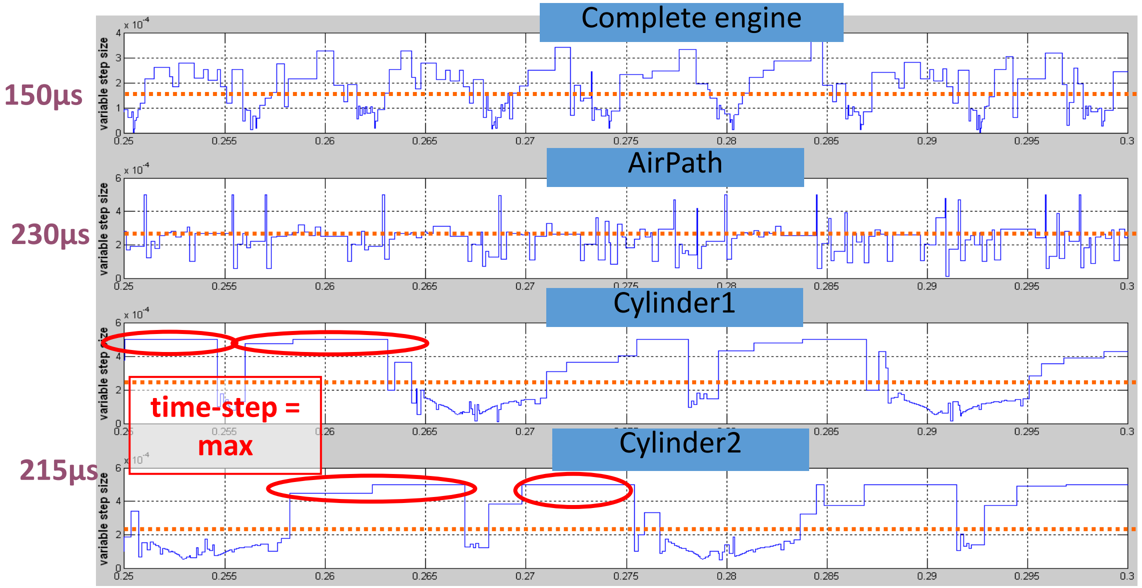
- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)
  - ➔ Mean value increased from **150µs** to **230µs**
- Result on speed-up
  - Mono-core simulation
    - 5 threads on 1 core
    - Speed-up ≈ **2**
    - Thanks to System splitting & Solver coupling
    - Despite multi threading cost
  - Multi-core simulation
    - 5 threads on 5 cores
    - Speed-up ≈ **8** (AP then 4Cyls in //)

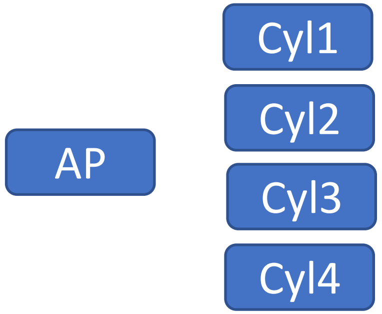




RESULTS FROM PREVIOUS WORK
SPLITTING IS SPEED-UP (CONT'D)

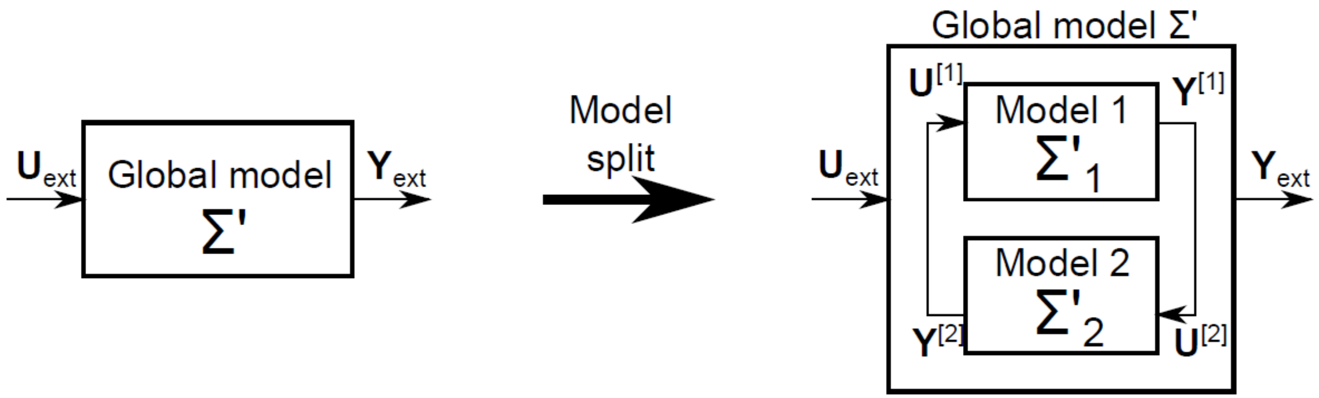
- Number of events is reduced locally
- Integration step can reach maximum allowed value (500µs)
    - ➔Mean value increased from **150µs** to **230µs**
- Result on speed-up
    - Mono-core simulation
        - 5 threads on 1 core
        - Speed-up ≈ **2**
        - Thanks to System splitting & Solver coupling
        - Despite multi threading cost
    - Multi-core simulation
        - 5 threads on 5 cores
        - Speed-up ≈ **8** (AP then 4Cyls in //)
        - Speed-up ≈ **9** (both AP and 4Cyls in //)





RESULTS FROM PREVIOUS WORK
IMPROVING PARALLELISM WITH THE RCOSIM APPROACH

- System splitting brings virtual algebraic loops



$$
\begin{aligned}
\dot{\boldsymbol{X}} &= \boldsymbol{f}(t, \boldsymbol{X}, \boldsymbol{U}_{ext}) \\
\boldsymbol{Y}_{ext} &= \boldsymbol{g}(t, \boldsymbol{X}, \boldsymbol{U}_{ext})
\end{aligned}
$$

$$
\begin{cases}
\dot{\boldsymbol{X}}^{[1]} = \boldsymbol{f}^{[1]}(t, \boldsymbol{X}^{[1]}, \boldsymbol{D}^{[1]}, \boldsymbol{U}^{[1]}, \boldsymbol{U}_{ext}) \\
\boldsymbol{Y}^{[1]} = \boldsymbol{g}^{[1]}(t, \boldsymbol{X}^{[1]}, \boldsymbol{D}^{[1]}, \boldsymbol{U}^{[1]}, \boldsymbol{U}_{ext})
\end{cases}
$$

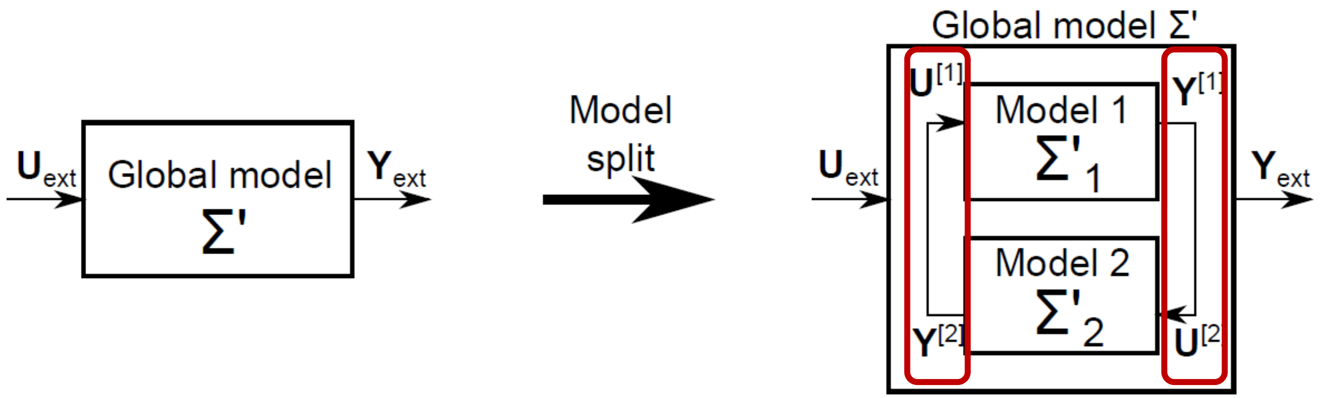
$$
\begin{cases}
\dot{\boldsymbol{X}}^{[2]} = \boldsymbol{f}^{[2]}(t, \boldsymbol{X}^{[2]}, \boldsymbol{D}^{[2]}, \boldsymbol{U}^{[2]}, \boldsymbol{U}_{ext}) \\
\boldsymbol{Y}^{[2]} = \boldsymbol{g}^{[2]}(t, \boldsymbol{X}^{[2]}, \boldsymbol{D}^{[2]}, \boldsymbol{U}^{[2]}, \boldsymbol{U}_{ext})
\end{cases}
$$

RESULTS FROM PREVIOUS WORK
IMPROVING PARALLELISM WITH THE RCOSIM APPROACH

- System splitting brings virtual algebraic loops

➔ Involve delayed outputs, even with an efficient execution order

➔ Problem with accuracy



$$
\begin{aligned}
\dot{X} &= f(t, X, U_{ext}) \\
Y_{ext} &= g(t, X, U_{ext})
\end{aligned}
$$

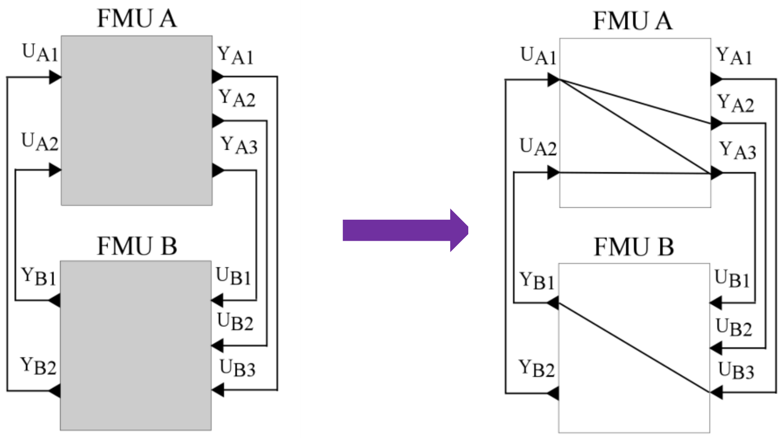
$$
\begin{cases}
\dot{X}^{[1]} = f^{[1]}(t, X^{[1]}, D^{[1]}, U^{[1]}, U_{ext}) \\
Y^{[1]} = g^{[1]}(t, X^{[1]}, D^{[1]}, U^{[1]}, U_{ext})
\end{cases}
$$

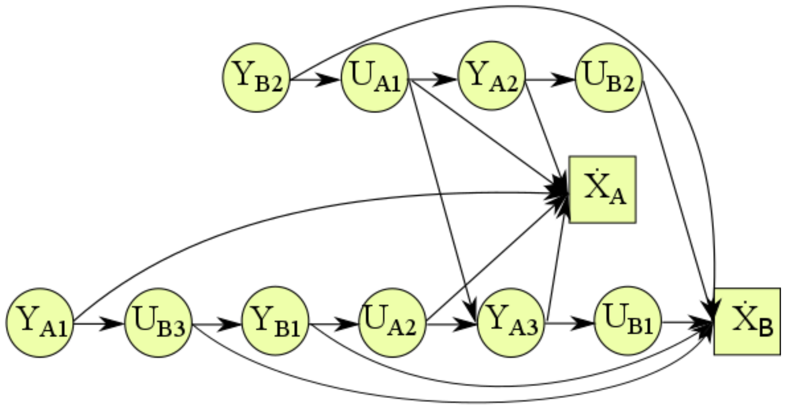
$$
\begin{cases}
\dot{X}^{[2]} = f^{[2]}(t, X^{[2]}, D^{[2]}, U^{[2]}, U_{ext}) \\
Y^{[2]} = g^{[2]}(t, X^{[2]}, D^{[2]}, U^{[2]}, U_{ext})
\end{cases}
$$

RESULTS FROM PREVIOUS WORK
RCOSIM: REFINED CO-SIMULATION

- Before FMI
  - Only dependencies between models are specified by the user
  - Models are black boxes → can't identify locally if Y is dependent on U
- With FMI
  - Relationships between each Y and U is known
  - Each Y and U is computed with a different FMU function

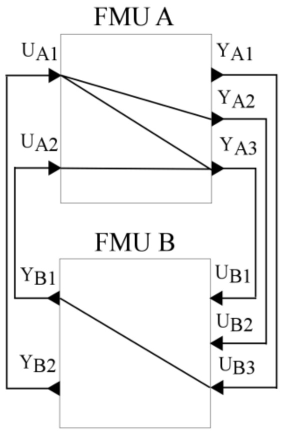
→ Build refined dependency graph
  - Vertices: IN, OUT and STATE operations
  - Directed edges: precedencies between operations
  - Target: Ordinary Differential Equations (ODEs)
    - No algebraic loops → Directed Acyclic Graph
- Apply a multi-core scheduling heuristic on the dataflow graph





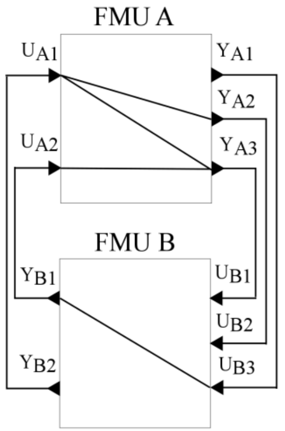
RESULTS FROM PREVIOUS WORK
IMPROVING PARALLELISM WITH THE RCOSIM APPROACH

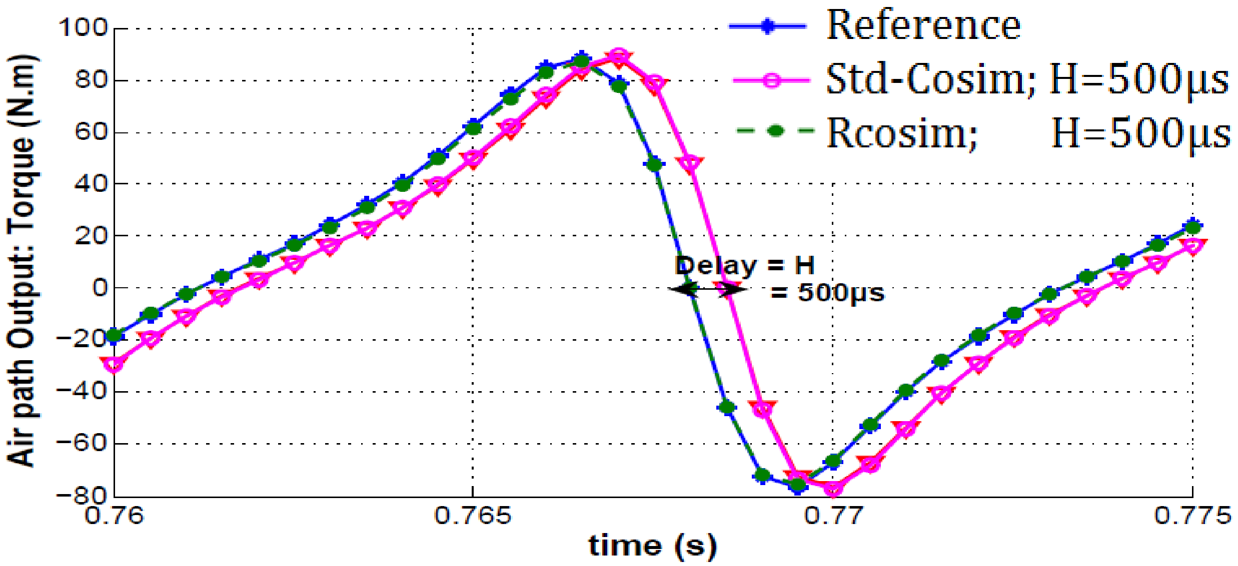
- Torque is a direct feedthrough output: e.g. Y_{A3}
- Expected delays with Standard Co-simulation (Std-Cosim) due to arbitrary order execution decision between models



RESULTS FROM PREVIOUS WORK
IMPROVING PARALLELISM WITH THE RCOSIM APPROACH

- 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
- Elimination of delays with RCosim
  - The execution order is compliant with initial model
- Speed-up ≈ **10**
  - No more delays → Correct data → Less iteration of the solver





Simulation method	Std-Cosim	RCosim
Er(%) with H=100µs	2.95	0.68
Er(%) with H=250µs	+572% 9.12	1.1
Er(%) with H=500µs	**19.83**	**1.37**

INSTITUT CARNOT IFPEN Transports Energie

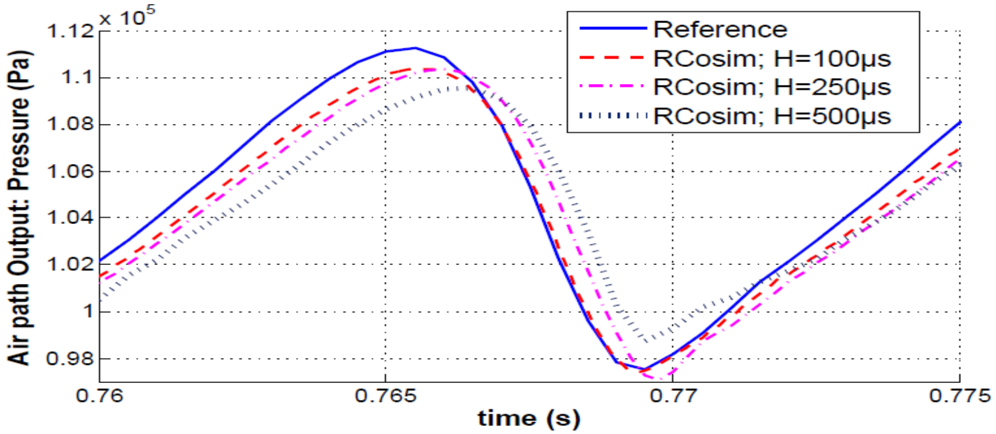
IFP Energies nouvelles

OUTLINE

- Background on co-simulation: context & challenges
- Results from previous work
- **Ensuring co-simulation accuracy with CHOPtrey extrapolation approach**
- Conclusion and perspectives

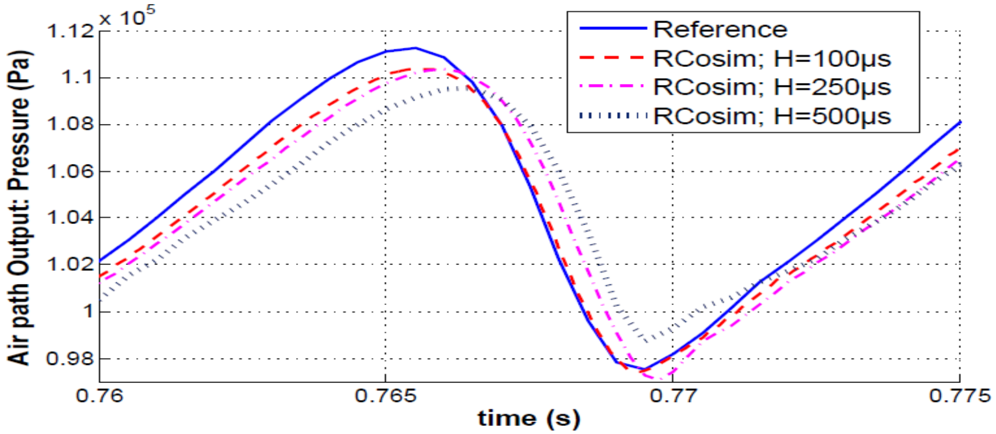
CHOPtrey EXTRAPOLATION APPROACH
IMPROVE AGAIN THE SIMULATION ACCURACY

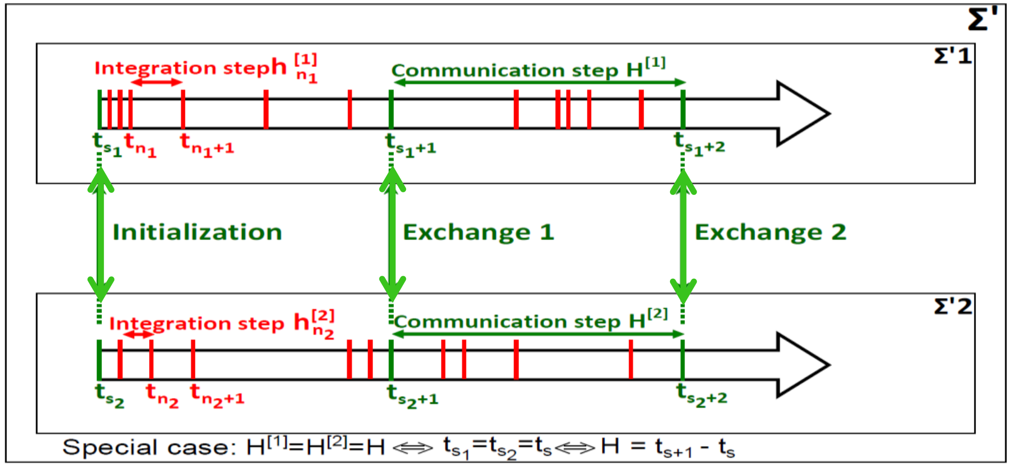
- Limitation: with RCosim, errors are reduced but still exist
- Reason: Input data is held constant during the communication step
- Dilemma:↗↗ communication step
  - ↗↗ Speed-up
  - ↗↗ Integration error



CHOPtrey EXTRAPOLATION APPROACH
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
  - ↗↗ Speed-up
  - ↗↗ Integration error
- Idea: Extrapolate input signals to
  - Enlarge intervals
  - Reduce simulation errors





RELATED WORK ON PREDICTION

- **Difficulties**
  - Related work on extrapolations treated mostly the continuous case
    - Successful for non-stiff systems
    - Encountered problems with stiff systems → polynomial prediction may fail
  - Complex systems with hybrid behavior is even more difficult to predict
    - Nonlinearities, 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 concept of context-based approach from lossless image encoders
  - Predict a pixel value based on a pattern of causal neighboring pixels
  - Different contexts: flat, smooth, +45° or -45° edges, etc.

INSTITUT CARNOT IFPEN Transports Energie

IFP Energies nouvelles

CHOPtrey EXTRAPOLATION APPROACH
A FAST AND CAUSAL PREDICTION

- We propose a Computationally Hasty Online Prediction framework (**CHOPred**)
- It is based on Causal Hopping Oblivious Polynomials (**CHOPoly**)
- $P_{\delta,\lambda,\omega}$: 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:

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

INSTITUT CARNOT IFPEN Transports Energie

Energies nouvelles

CHOPtrey EXTRAPOLATION APPROACH
A FAST AND CAUSAL PREDICTION

- We propose a Computationally Hasty Online Prediction framework (**CHOPred**)
- It is based on Causal Hopping Oblivious Polynomials (**CHOPoly**)
- $P_{\delta,\lambda,\omega}$: 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: $\overline{z}_{d,\lambda,\omega} = \sum_{l=0}^{\lambda-1} (\lambda - l)^{\omega} l^d$

Use of LUT ➔ Fast computation

- General formula for extrapolation:

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

CHOPtrey EXTRAPOLATION APPROACH
A FAST AND CAUSAL PREDICTION

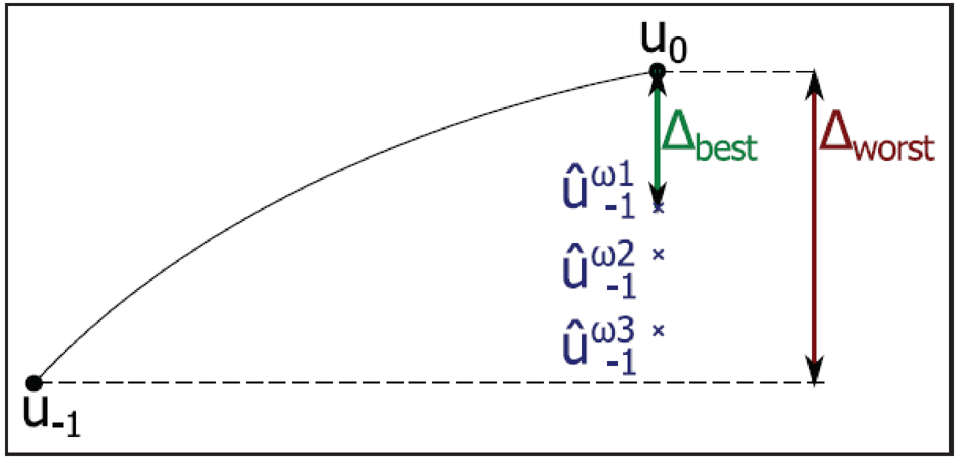
- We propose a Computationally Hasty Online Prediction framework (**CHOPred**)
- It is based on Causal Hopping Oblivious Polynomials (**CHOPoly**)
- $P_{\delta,\lambda,\omega}$: least squares polynomial predictor
  - δ: prediction degree;
  - λ: prediction frame length;
  - ω: weighting factor
- u: input 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:

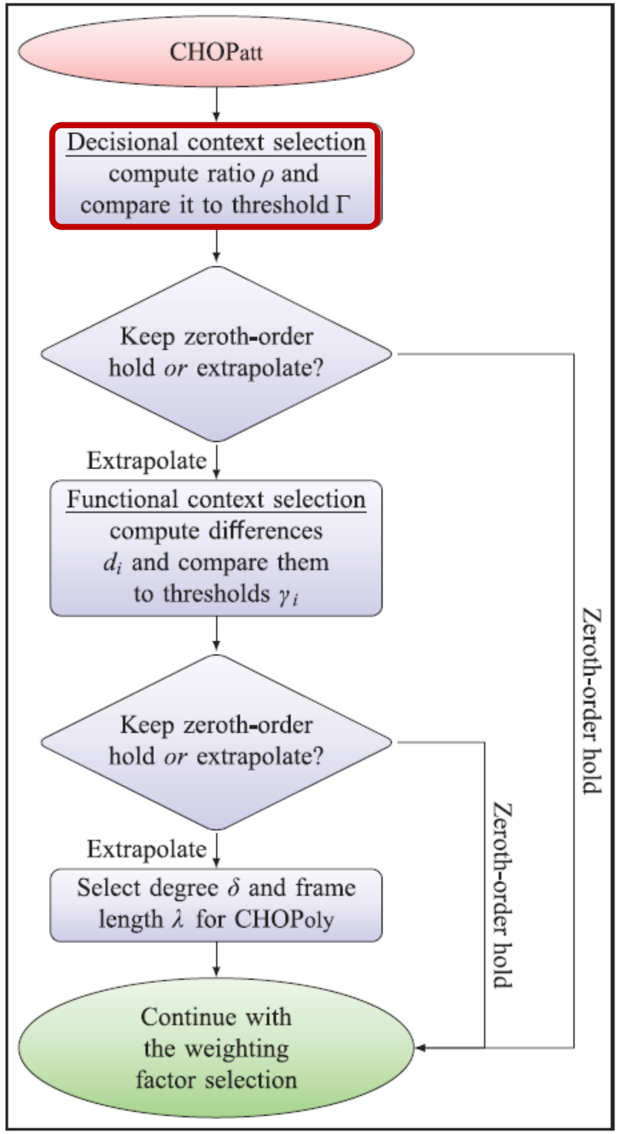
Computed at communication steps only

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

CHOPtrey EXTRAPOLATION APPROACH
A RELIABLE PREDICTION

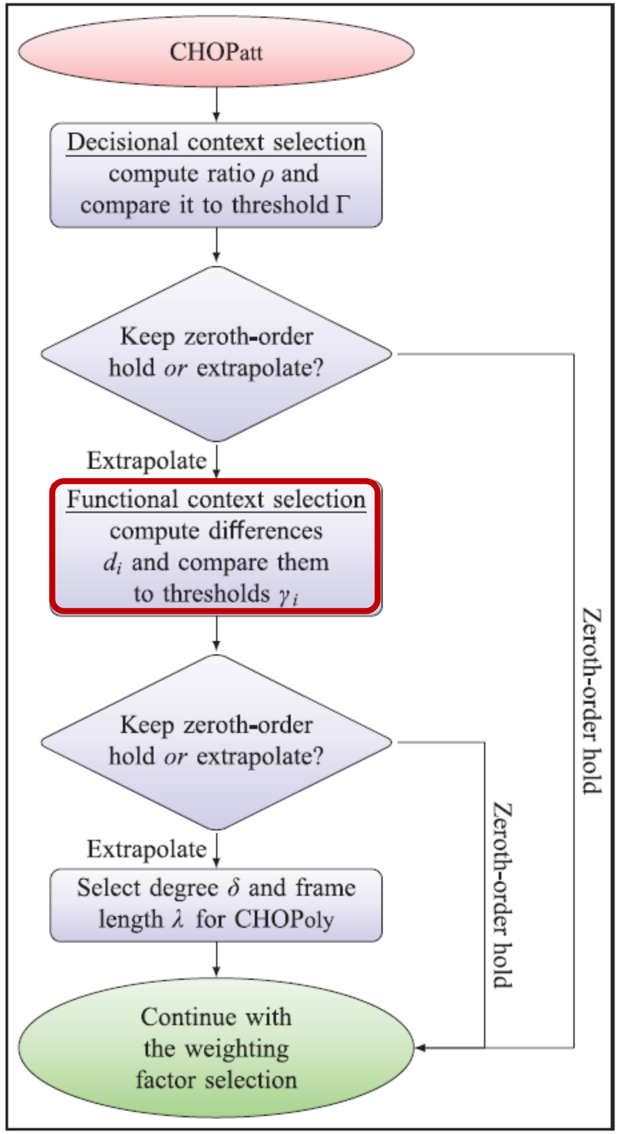
- It uses a Contextual & Hierarchical Ontology of Patterns (**CHOPatt**)
  - To handle the discontinuities by selecting the appropriate $P_{\delta,\lambda,\omega}$
- **STEP1:** Decisional context selection
  - Worst case scenario without extrapolation: $\Delta_{\text{worst}} = |u_0 - u_{-1}|$
  - Best prediction pattern: $\Delta_{\text{best}} = \min_{\omega\in\Omega} |u_0 - \hat{u}^{\omega}_{-1}|\ ;\quad \Omega = \{0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$
  - 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 "cliff" context





CHOPtrey EXTRAPOLATION APPROACH
A RELIABLE PREDICTION

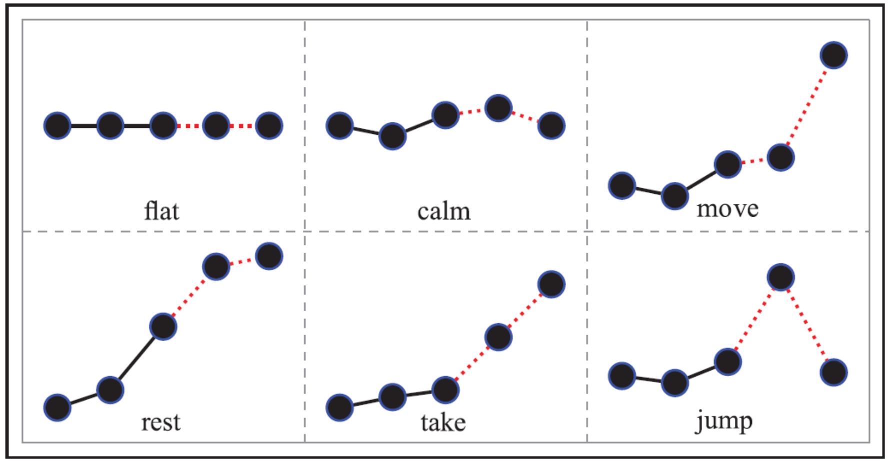
- **STEP2:** Functional context selection
  - Differences (variations): $d_0 = u_0 - u_{-1}$ and $d_{-1} = u_{-1} - u_{-2}$
  - Thresholds: $\gamma_0 = \gamma_{-1} = \frac{1}{2} \max_{i \in [1-\Lambda, \ldots, -3]} (|u_i - u_{i+1}|)$
  - Conditions:
    - O if $|d_i| = 0$;
    - C_i if $0 < |d_i| \leqslant \gamma_i$;
    - $\overline{C_i}$ if $|d_i| > \gamma_i$.

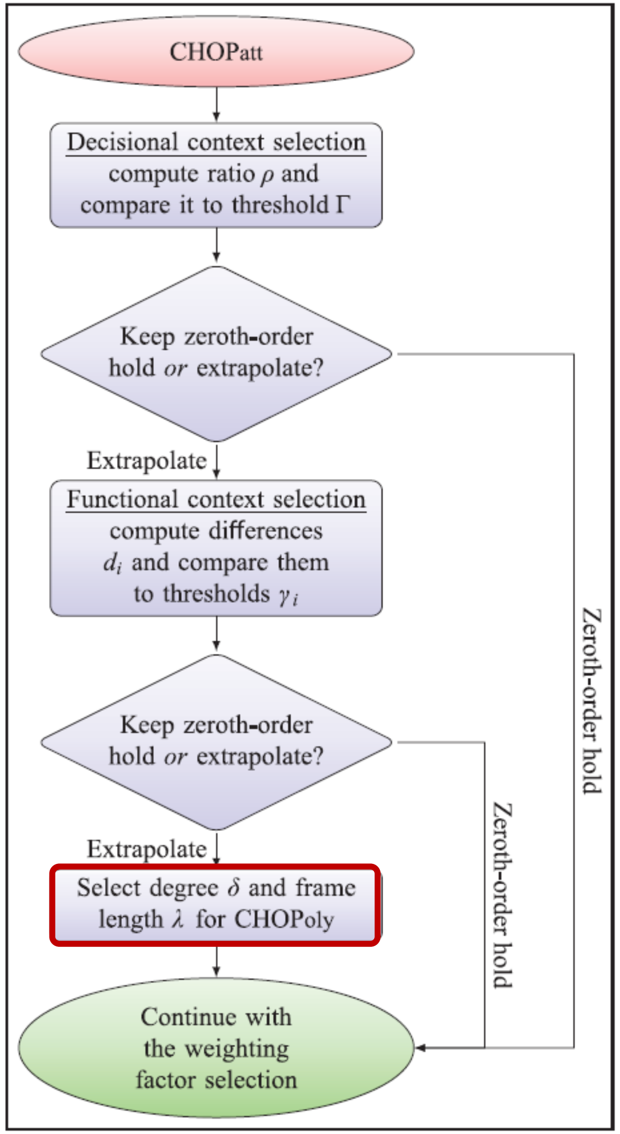


CHOPtrey EXTRAPOLATION APPROACH
A RELIABLE PREDICTION

- **STEP2:** 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

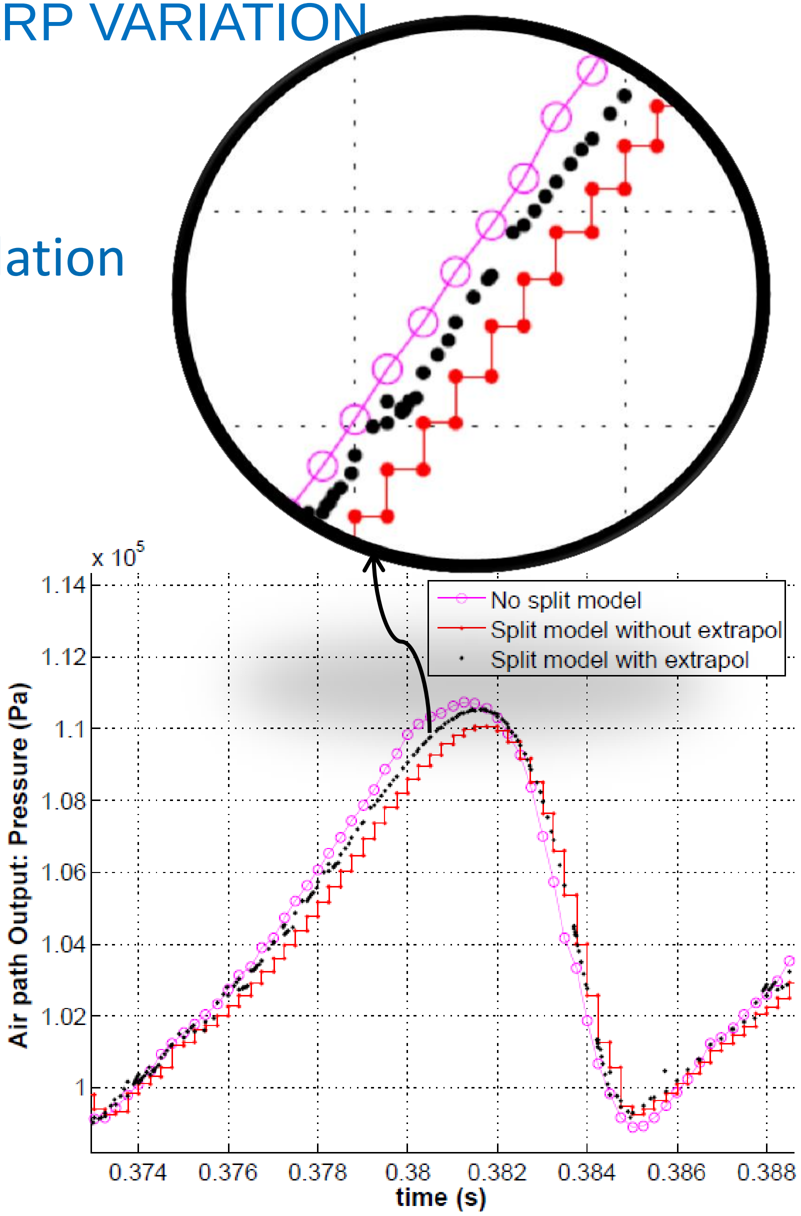
- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

➔ CHOPtrey?

SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC DETECTION OF SHARP VARIATION

- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

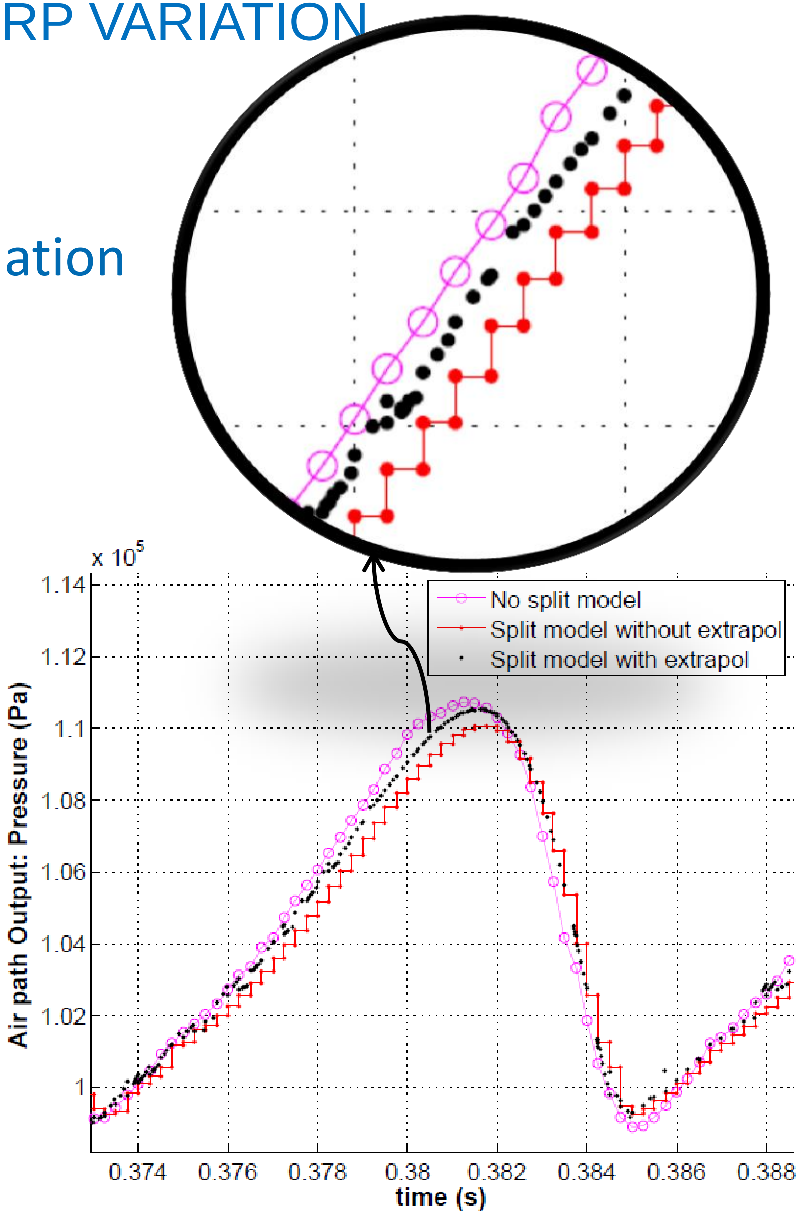
➔ CHOPtrey?

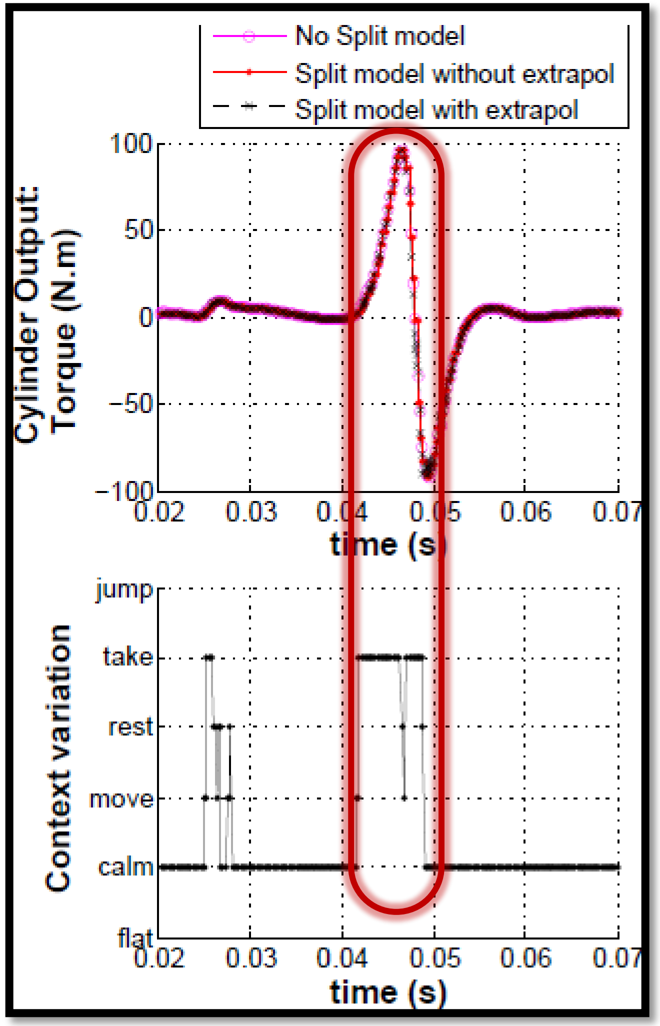


SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC DETECTION OF SHARP VARIATION

- Conventional 1st & 2nd order extrapolation
  - Fails on the engine model
  - Major causes:
    - Discontinuities
    - Sharp variations

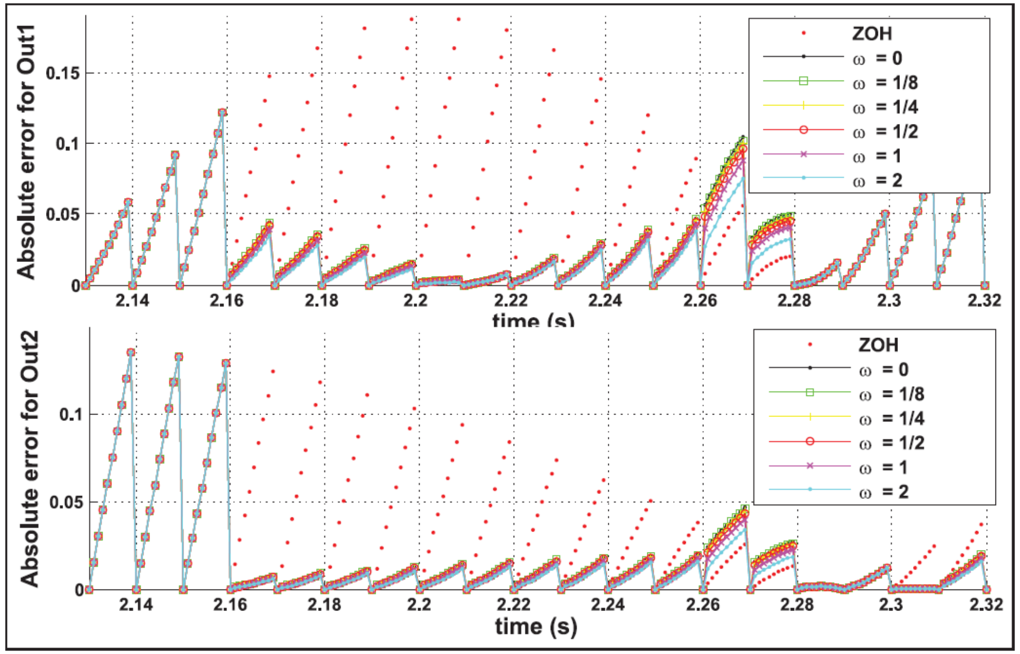
➔ CHOPtrey?





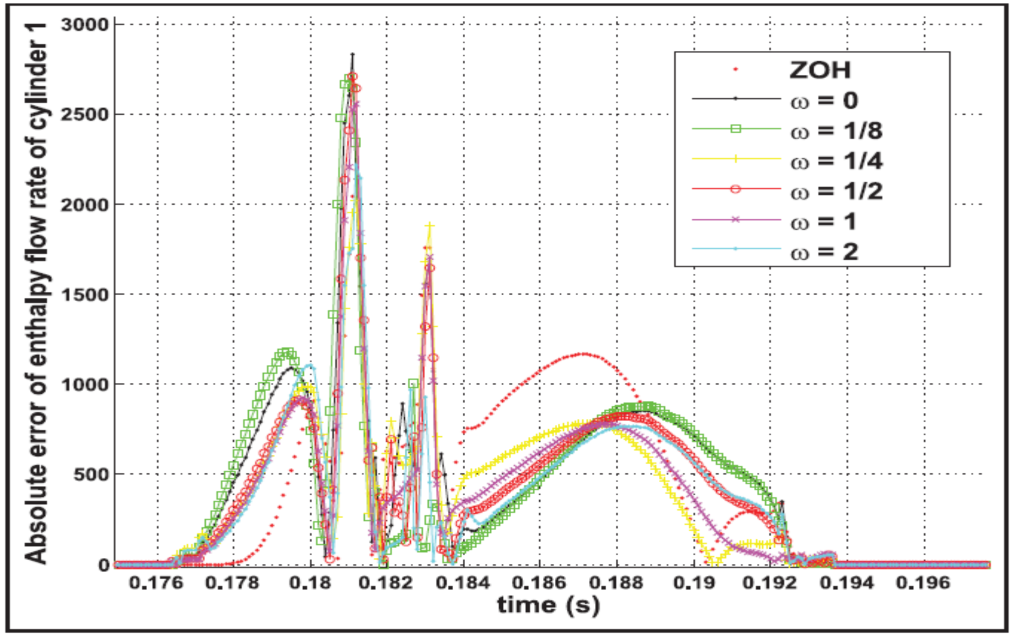
SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC SELECTION OF THE WEIGHTING FACTOR

- Simple model with no coupling
  - ➔ The higher the weighting factor, the smaller the error



SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC SELECTION OF THE WEIGHTING FACTOR

- Simple model with no coupling
  - ➔ The higher the weighting factor, the smaller the error
- Complex coupled models, i.e. engine model
  - ➔ No unique best weighting factor ω

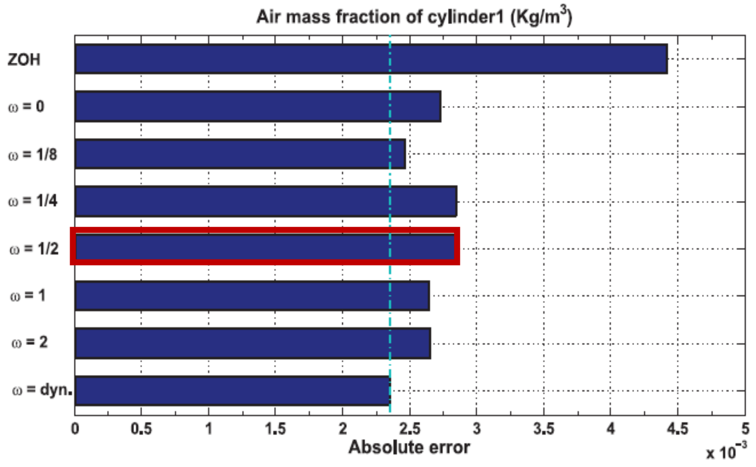


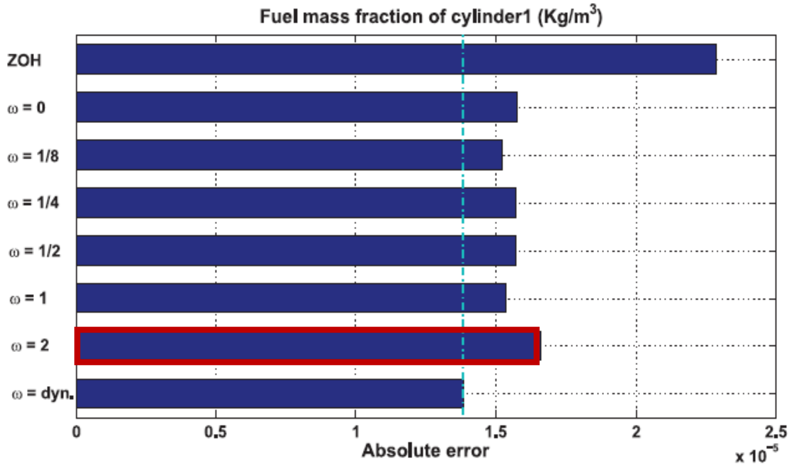
SIMULATION RESULTS WITH CHOPtrey
AUTOMATIC SELECTION OF THE WEIGHTING FACTOR

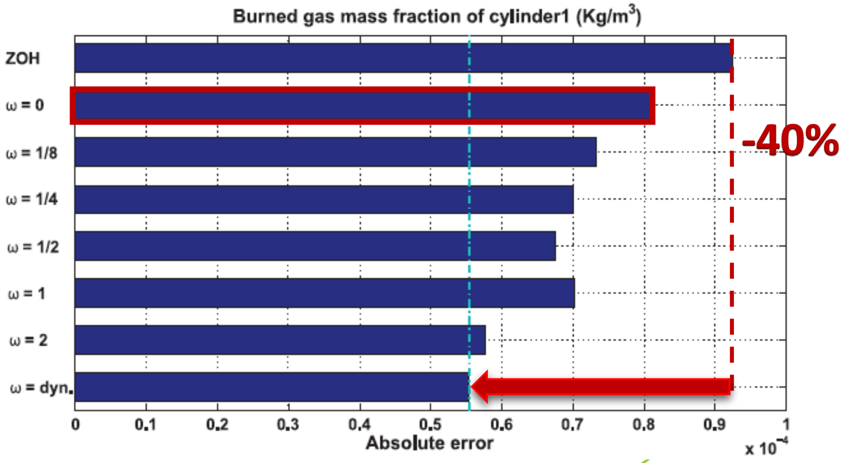
- Simple model with no coupling
  - ➔ The higher the weighting factor, the smaller the error
- Complex coupled models, i.e. engine model
  - ➔ No unique best weighting factor ω

➔ 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 still 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	7	341
	CHOPtrey	10.07	-26	21

+12.5%

OUTLINE

- Background on co-simulation: context & challenges
- Results from previous work
- Ensuring co-simulation accuracy with CHOPtrey extrapolation approach
- **Conclusion and perspectives**

CONCLUSION

- The use of large communication steps allows to accelerate the simulation at the cost of precision
- Conventional extrapolation methods fails with hybrid dynamical systems

➔ CHOPtrey extrapolation technique provides a solution for the trade-off between speed-up and accuracy, thanks to

  - The combination of a prediction and a multi-level context selection
  - Negligible computational overheads

- CHOPtrey combination with model splitting and parallel simulation on a hybrid dynamical engine model allows supra-linear speed-up (10 time faster with 5 cores) with acceptable result accuracy

PERSPECTIVES

- Decompose signals into morphological components such as polynomial trends, singularities and oscillations
  - Allow to adapt detection thresholds
  - ➔ Improve context assignment
- Use of the knowledge of the plant model
  - Discard out-of-bound values as nonnegative variables
  - ➔ Improve the discrimination of cliff behaviors
- Use of adaptive communication steps
  - Context-based and error-based closed-loop control
- Access on the input derivatives of the models
  - Provided by FMI for co-simulation
  - ➔ Improve the extrapolation

Innovating for energy

Find us on:

www.ifpenergiesnouvelles.com

@IFPENinnovation