ITMO UNIVERSITY

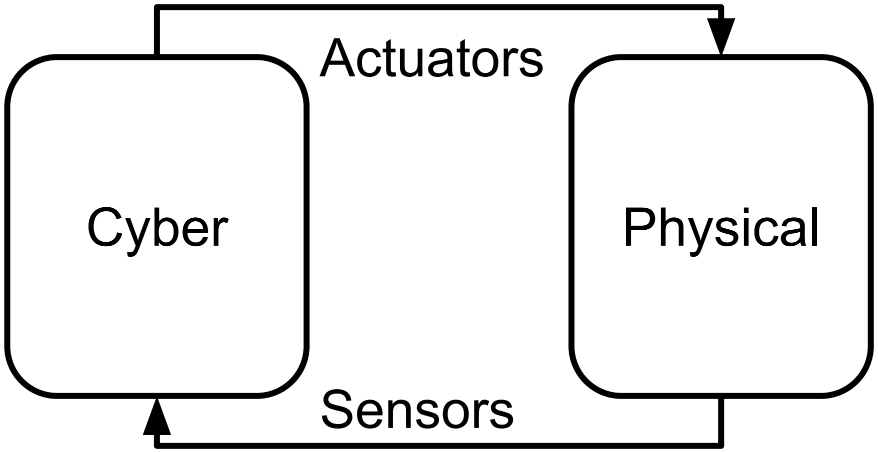
Plant trace generation for formal plant model inference: methods and case study

Dmitry Avdyukhin, Daniil Chivilikhin, Georgiy Korneev, Vladimir Ulyantsev, Anatoly Shalyto

INDIN’2017

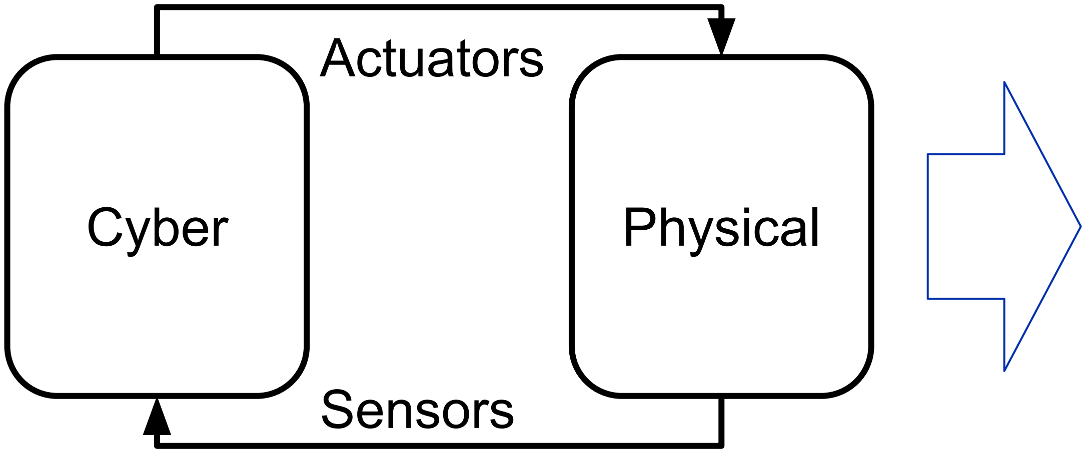
Emden, Germany, 25 July 2017

Cyber-physical systems correctness



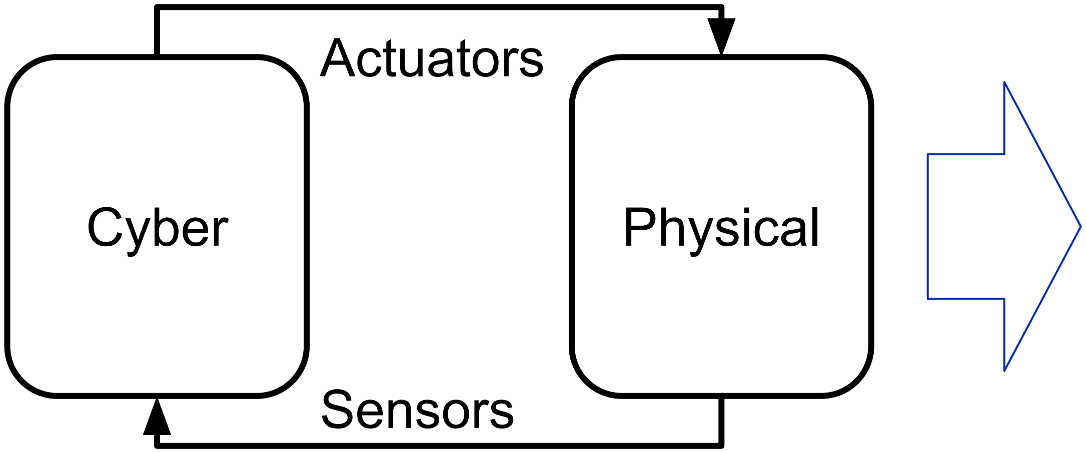
https://i.ytimg.com/vi/zhBxfNm3rJE/maxresdefault.jpg

Cyber-physical systems correctness



https://i.ytimg.com/vi/zhBxfNm3rJE/maxresdefault.jpg

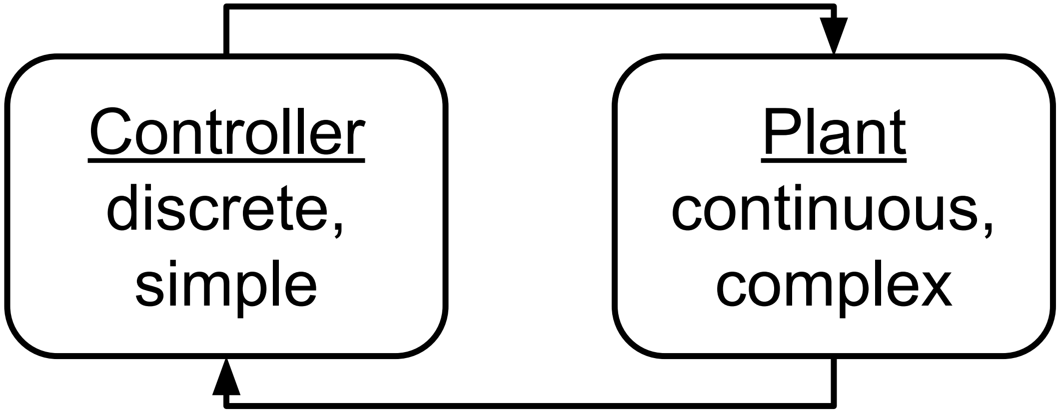
Cyber-physical systems correctness



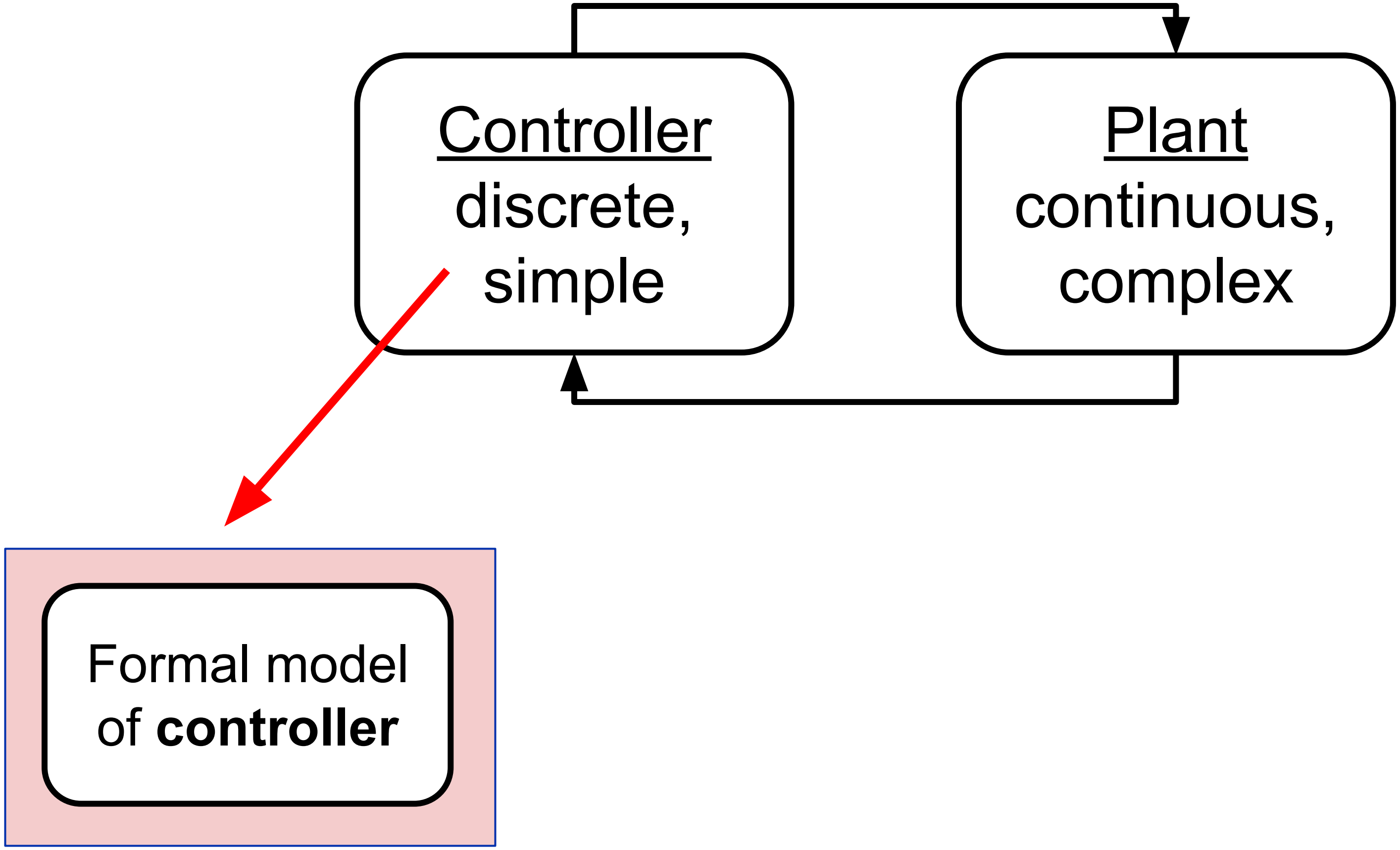
Testing + Simulation + **Verification**

https://i.ytimg.com/vi/zhBxfNm3rJE/maxresdefault.jpg

CPS verification

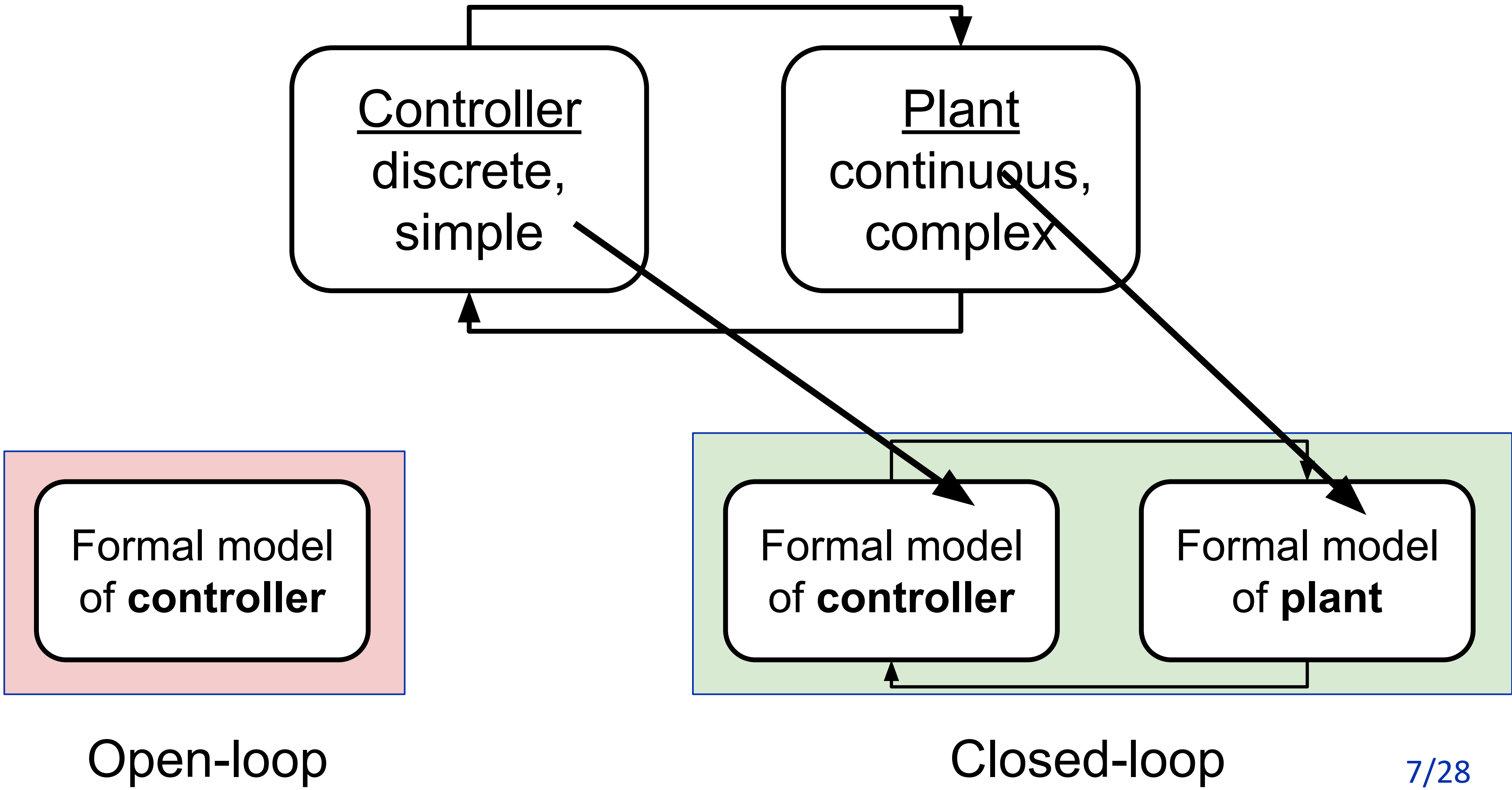


CPS verification



Open-loop

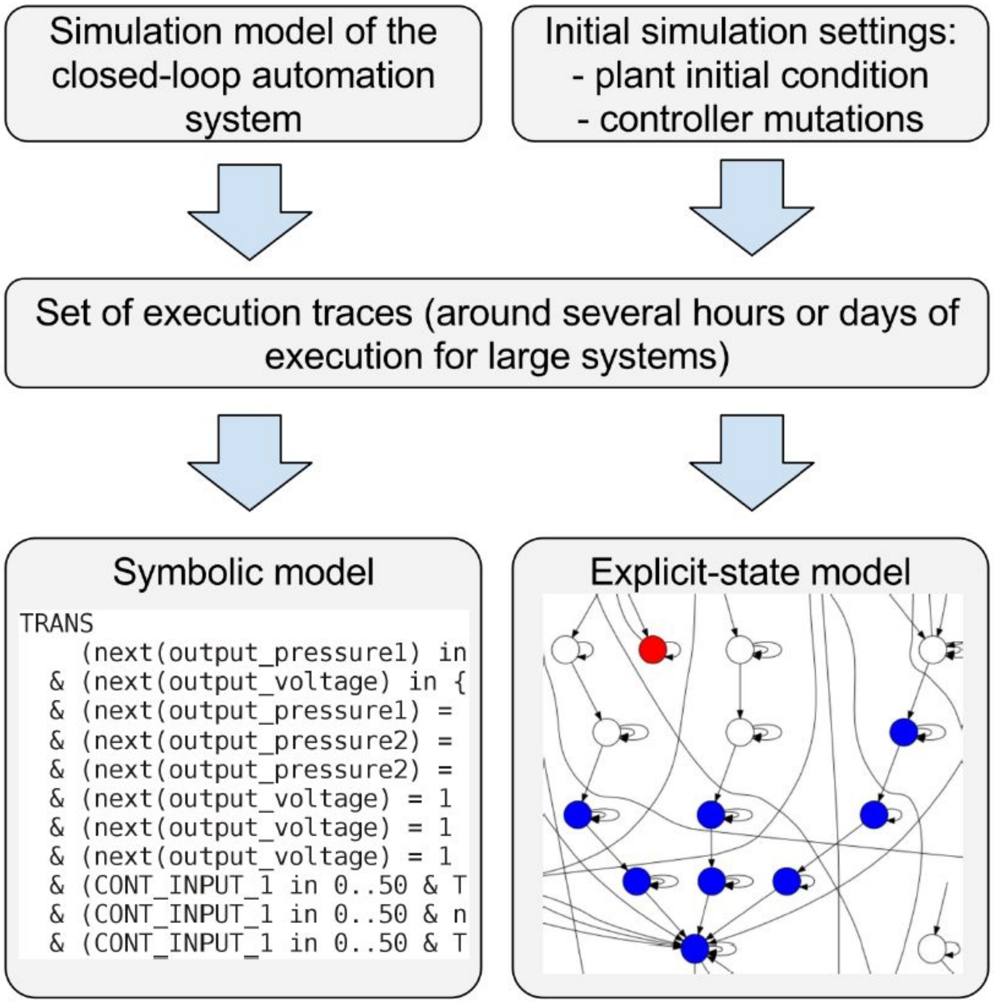
CPS verification



CPS verification

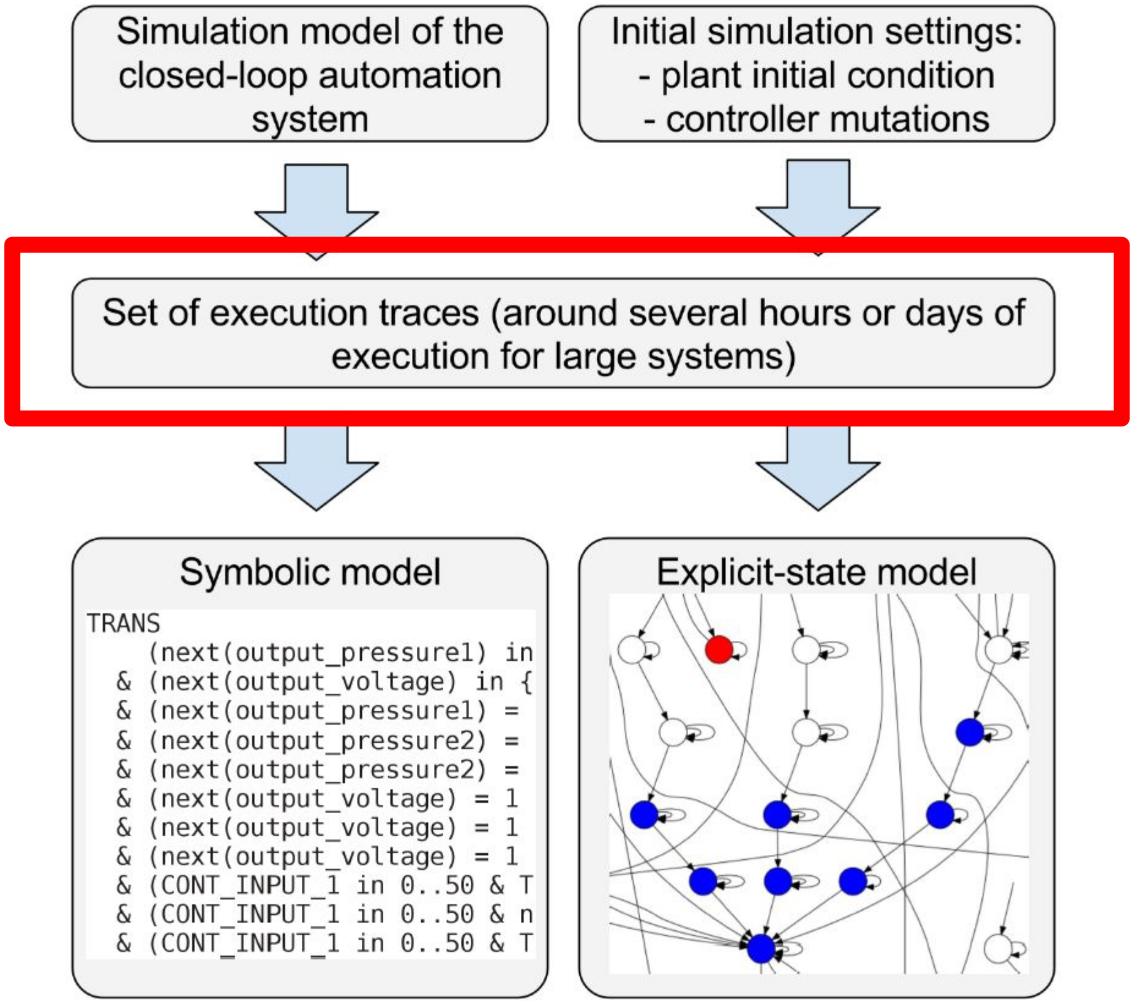
Open-loop: controller only	Closed-loop: plant + controller
• Restriction on checked properties • State explosion • Often “incorrect” results because of unrealistic input	• Correctness of the entire system • **Requires plant model**

Automatic plant model inference



[Buzhinsky, Vyatkin / INDIN'16]

Automatic plant model inference



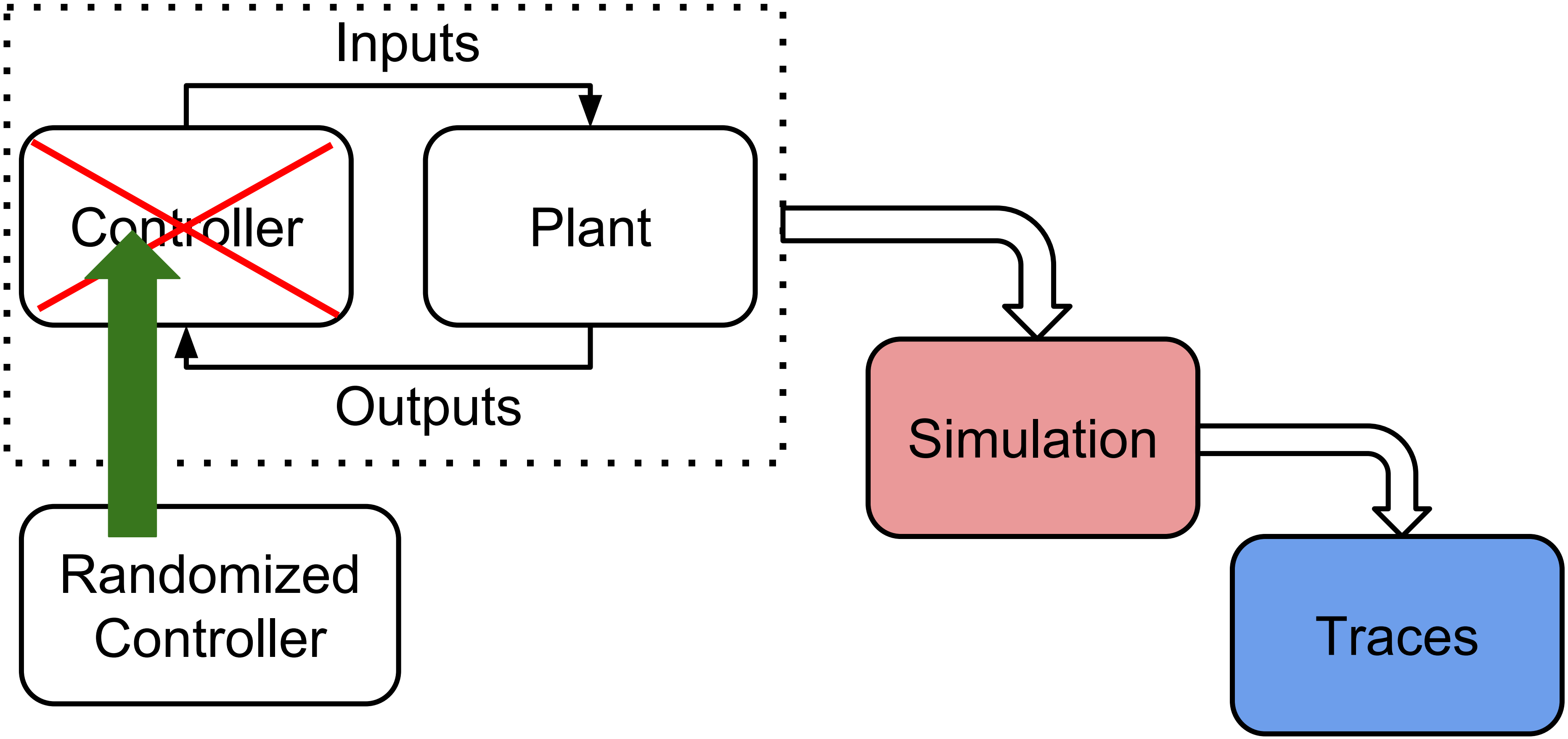
?

[Buzhinsky, Vyatkin / INDIN’16]

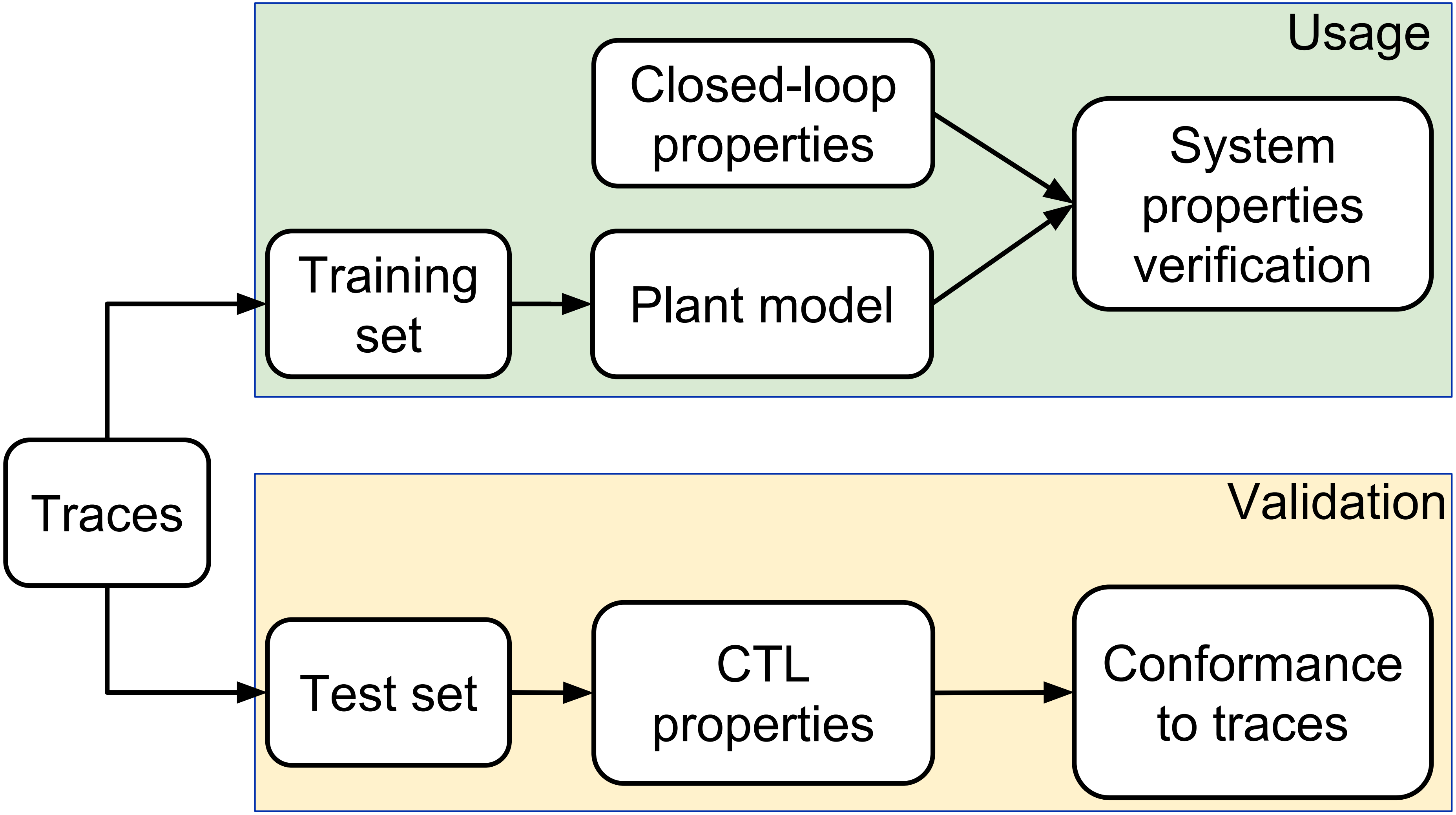
Goals

- Propose and analyze plant **trace generation methods**
  - Applicable to wide range of systems
  - Provide good coverage of plant behavior

Pipeline (1)


Inputs
Controller
Plant
Outputs
Simulation
Traces
Randomized Controller

Pipeline (2)

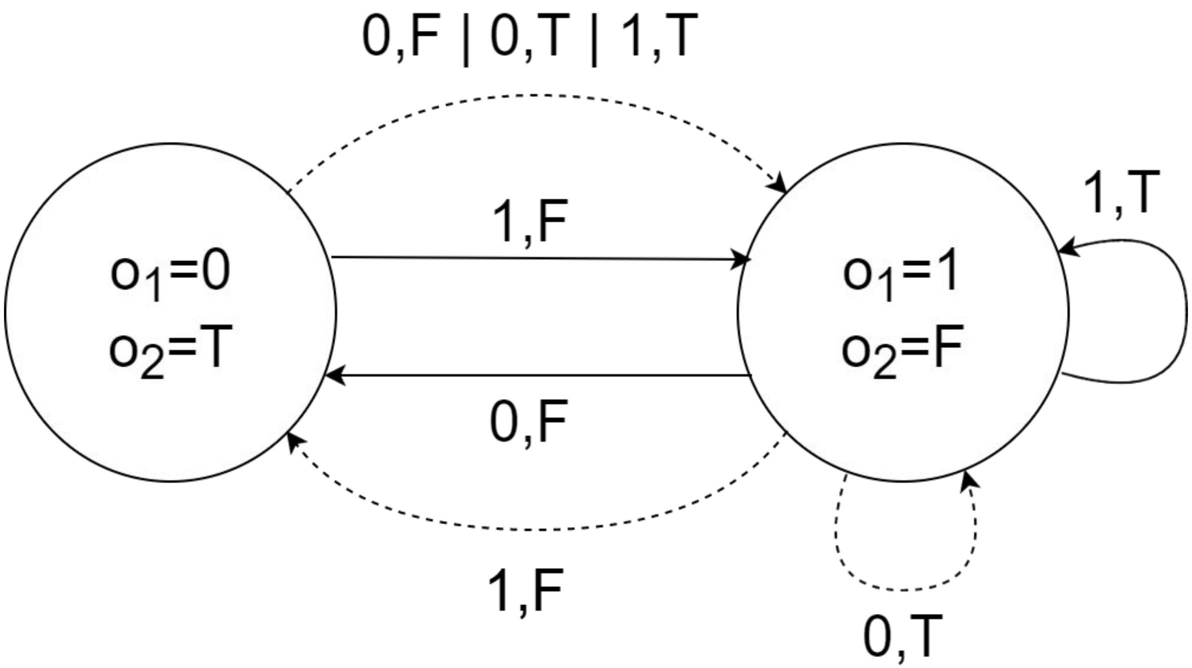

Usage
Closed-loop properties
System properties verification
Training set
Plant model
Traces
Validation
Test set
CTL properties
Conformance to traces

Plant model generation from traces

- Moore machine
  - Transition labels are different input combinations
  - At most one state for each output combination
- Discretization
  - $[0; 100] \longrightarrow \{0\} \cup (0; 100) \cup \{100\}$

Explicit-state plant model generation

- Only states and transitions encountered in traces
- “Unsupported” transitions to accept all inputs



Constraint-based plant model generation

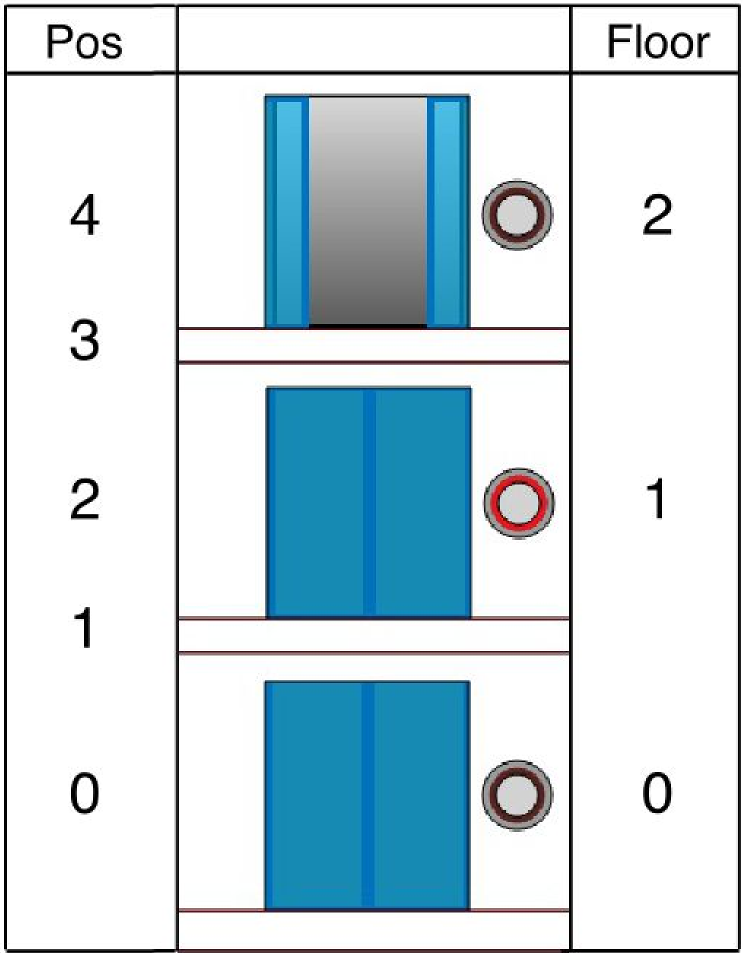
- Variable for each input and output
- Each pair of variables can only have values found in traces
  - $o_1=0 \wedge o_2=T$
  - $o_2=T \longrightarrow next(o_2)=F$
  - $i_1=0 \longrightarrow next(o_1)=0$
- Changeability constraint
  - “Some output will eventually change”
  - To avoid eternal loops

Proposed trace generation methods

1. Random controller
   a. Generate random inputs each cycle
2. “Semirandom(***C***)” controller
   a. Generate random inputs and do not change them until ***C*** cycles pass or some output changes
   b. Allows to visit rare states
3. Uniform inputs coverage
   a. The probability to take a certain value is inversely proportional to its frequency in traces

Case study: elevator

- **Inputs**
  - Up, Down
  - OpenDoor0..2
- **Outputs**
  - Button0..2 ∈ {0, 1}
  - Floor0..2 ∈ {0, 1}
  - Closed0..2 ∈ {0, 1}
  - Position ∈ ℝ



Model conformance to traces

- Is the trace accepted by the model?
  - Is the model general enough?

- $(O_1, I_1), (O_1, I_2), \ldots, (O_n, I_n)$

$\rightarrow$ **EF**$(O_1 \wedge I_1 \wedge$ **EX**$(O_2 \wedge I_2 \wedge$ **EX**$(O_3 \wedge \ldots)))$

- Cross-check conformance to traces generated by different methods

Conformance to traces

- Training set is always accepted

Conformance to traces

- Training set is always accepted

Trace generation method	Model							
	Original		Random		Semirandom(10)		Semirandom(100)	
	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit
original	100 %	99.96 %	31 %	14 %	76.3 %	14 %	100 %	24 %
random	0 %	0 %	100 %	98.7 %	100 %	96.5 %	100 %	79 %
semirandom(10)	0.07 %	0.07 %	99.9 %	94 %	100 %	96 %	100 %	93.5 %
semirandom(100)	1.3 %	2.5 %	67 %	63 %	99.4 %	82 %	100 %	93.5 %

- Original model is not general enough

ITMO UNIVERSITY

Conformance to traces

- Training set is always accepted

Trace generation method	Model							
	Original		Random		Semirandom(10)		Semirandom(100)	
	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit
original	100 %	99.96 %	31 %	14 %	76.3 %	14 %	100 %	24 %
random	0 %	0 %	100 %	98.7 %	100 %	96.5 %	100 %	79 %
semirandom(10)	0.07 %	0.07 %	99.9 %	94 %	100 %	96 %	100 %	93.5 %
semirandom(100)	1.3 %	2.5 %	67 %	63 %	99.4 %	82 %	100 %	93.5 %

- Original model is not general enough
- Semirandom(100) > Semirandom(10) > Random = Semirandom(1)

Conformance to traces

- Training set is always accepted

Trace generation method	Model							
	Original		Random		Semirandom(10)		Semirandom(100)	
	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit	Constraint	Explicit
original	100 %	99.96 %	31 %	14 %	76.3 %	14 %	100 %	24 %
random	0 %	0 %	100 %	98.7 %	100 %	96.5 %	100 %	79 %
semirandom(10)	0.07 %	0.07 %	99.9 %	94 %	100 %	96 %	100 %	93.5 %
semirandom(100)	1.3 %	2.5 %	67 %	63 %	99.4 %	82 %	100 %	93.5 %

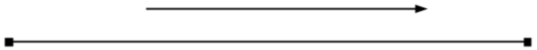
- Original model is not general enough
- Semirandom(100) > Semirandom(10) > Random = Semirandom(1)
- Explicit-state models – never 100%

System properties verification

Property	Meaning	Corr	Calc
$G(Floor1 \land G \neg Up \land G(Down \lor Floor0) \rightarrow G \neg Floor2)$	If the car is on the first floor and never moves up and always moves down or stays on floor 0, it will never reach floor 2	+	+
$G(Pos = 4 \land Down \land \neg Up \rightarrow X\ Pos = 3)$	If the car stays on floor 2 and moves down, it will be between floors 1 and 2	+	+

φ_1	$\mathbf{G}(\mathtt{Floor1} \wedge \mathbf{G}\, \neg\mathtt{Up} \wedge \mathbf{G}(\mathtt{Down} \vee \mathtt{Floor0}) \to \mathbf{G}\, \neg\mathtt{Floor2})$
φ_2	$\mathbf{G}(\mathbf{G}\, \neg\mathtt{Up} \wedge \mathbf{G}(\mathtt{Down} \vee \mathtt{Floor0}) \to \mathbf{F}\, \mathtt{Floor0})$
φ_3	$\mathbf{G}(\mathbf{G}\, \neg\mathtt{Down} \wedge \mathbf{G}(\mathtt{Up} \vee \mathtt{Floor2}) \to \mathbf{F}\, \mathtt{Floor2})$
φ_4	$\mathbf{G}\, \mathbf{F}\, \neg\mathtt{Down}$
φ_5	$\mathbf{G}(\mathtt{Floor1} \wedge \mathbf{G}\, \neg\mathtt{Up} \wedge \mathbf{G}\, \mathtt{Down} \to \mathbf{F}\, \mathtt{Floor2})$
φ_6	$\mathbf{G}\, \neg(\mathtt{Down} \wedge \mathtt{Up})$
φ_7	$\mathbf{G}(\mathtt{Pos} = 4 \wedge \mathtt{Down} \wedge \neg\mathtt{Up} \to \mathbf{X}\, \mathtt{Pos} = 3)$
φ_8	$\mathbf{G}(\mathtt{Pos} = 2 \wedge \mathtt{Down} \wedge \neg\mathtt{Up} \to \mathbf{X}\, \mathtt{Pos} = 1)$
φ_9	$\mathbf{G}(\mathtt{Pos} = 2 \wedge \neg\mathtt{Down} \wedge \mathtt{Up} \to \mathbf{X}\, \mathtt{Pos} = 3)$
φ_{10}	$\mathbf{G}(\mathtt{Pos} = 0 \wedge \neg\mathtt{Down} \wedge \mathtt{Up} \to \mathbf{X}\, \mathtt{Pos} = 1)$
φ_{11}	$\forall k \in [0..2]\ \mathbf{G}(\mathtt{Button}_k \to \mathbf{F}\, \mathtt{Floor}_k)$
φ_{12}	$\mathbf{G}(\mathtt{Button2} \wedge \text{(not always at some floor)} \to \mathbf{F}\, \mathtt{Floor2})$
φ_{13}	$\mathbf{G}(\mathtt{Button1} \wedge \text{(not always at some floor)} \to \mathbf{F}\, \mathtt{Floor1})$
φ_{14}	$\mathbf{G}(\mathtt{Button0} \wedge \text{(not always at some floor)} \to \mathbf{F}\, \mathtt{Floor0})$
φ_{15}	$\mathbf{G}(\mathtt{Pos} \in \{1,3\} \to \mathtt{DoorClosed0} \wedge \mathtt{DoorClosed1} \wedge \mathtt{DoorClosed2})$

Modifications of constraint-based plant generation method

- Constraints of form $O_i \wedge I_j \longrightarrow next(O_i)$
- Grouping of related inputs, such as (Up, Down)
- Changeability constraint
  - **G F** ¬Down is true – restriction on inputs
  - Some correct behavior is prohibited
  - Solution: 
    - Will eventually reach the end
    - When output depends on single input
    - Special case: if $i_k = v$ then o_j increases

Verification: different trace generation methods

- Original: does not allow unusual behavior
- Random, Semirandom and Uniform – similar results
- Explicit state violates some properties
  - Unsupported transitions are bad
- Constraint-based: after proposed modifications all verification results are correct

Conclusion

- Trace generation methods are proposed
  - Random – does not reach rare states
  - **Semirandom – good results**
  - Uniform – not different from semirandom
- Plant model generation methods modification
  - Constraints of form $O_i \wedge I_j \rightarrow next(O_i)$
  - Input grouping
  - Additional fairness constraints

ITMO UNIVERSITY

Thank you for your attention!

chivdan@rain.ifmo.ru

Acknowledgements

This work was supported by the Ministry of Education and Science of the Russian Federation, project RFMEFI58716X0032

Emden, Germany, 2017