Trends in formal verification in the railway signalling domain

Alessandro Fantechi[1,2], with contributions from:
Alessio Ferrari[2] Stefania Gnesi [2]
Gianluca Magnani[3]
Daniele Grasso[3]

[1]University of Florence, D.S.I., Florence, Italy

[2]ISTI - CNR, Pisa, Italy

[3]General Electric Transportation Systems (GETS), Firenze, Italy

February 20, 2012

Outline

Some views from experiences in applications of model checking to railway signalling equipments

Railway Signalling equipments

Traditional classification of major systems in:

- *train control systems*, that guarantee safe speed and braking control for trains, along the line
- *interlocking systems*, that establish safe routes through the intricate layout of tracks and points of a station.

$\longrightarrow$ Trend towards greater integration of the two classes , with equipments providing both types of functions.
However, maintaining this classification is useful to understand the major challenges from the verification point of view. -

Safety of Railway Signalling equipments

- in train control systems, the main safety criterion is to guarantee that two trains travelling at speed in the same direction stay a safe distance apart.
- in interlocking systems, safety is ensured when two trains cannot collide due to attempted occupancy of the same track or point. Other safety issues : e.g. *No Derailing property*: While a train is crossing a point, the point shall not change its position.

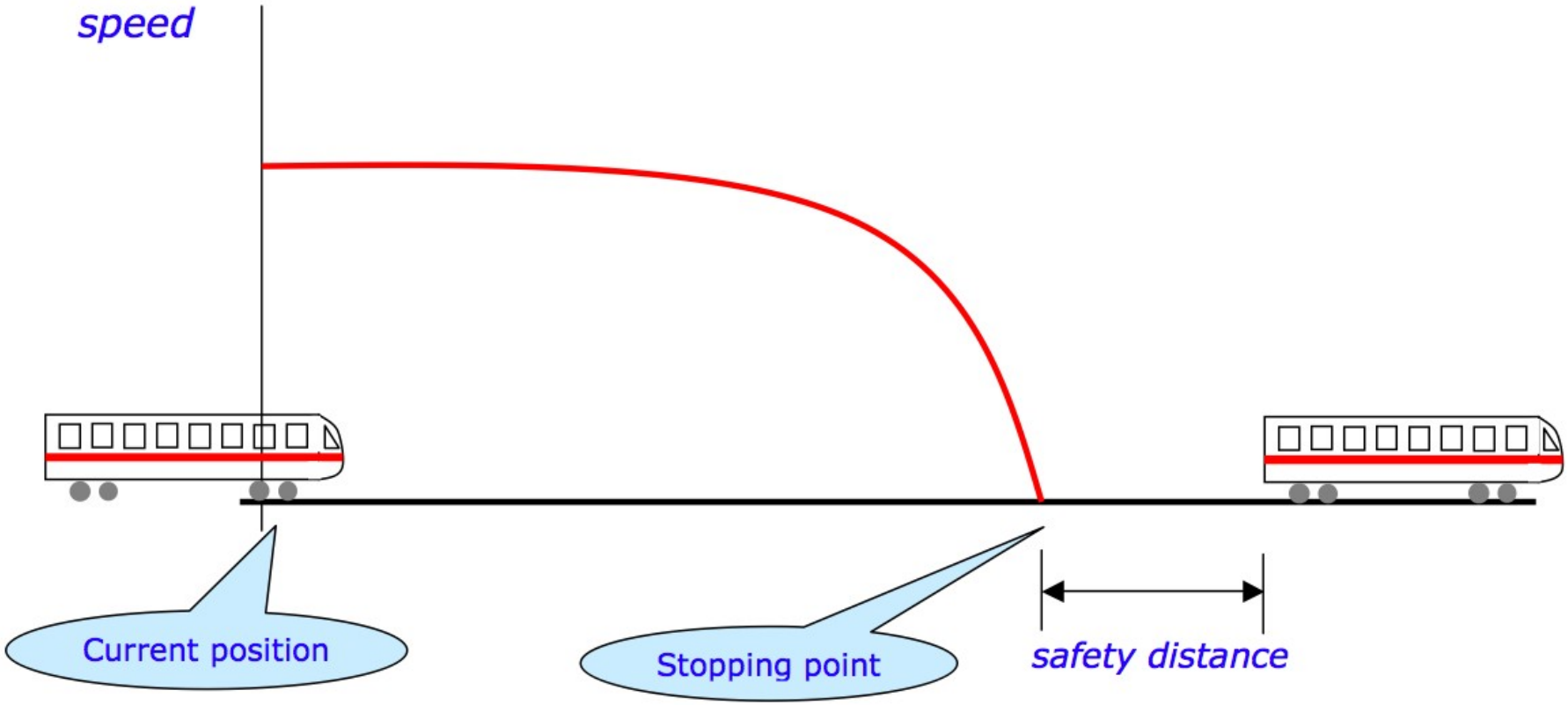
Train control equipments

- ATP - Automatic Train Protection
  The equipment protects the train from crashing into preceding train + train integrity control
- ATC - Automatic Train Control
  The equipment controls whether the operation of the driver always stays in the limits dictated by the signalling and by the line topology
- ATO - Automatic Train Operation
  The equipment operates the train. (UTO - Unmanned Train Operation if there is no human supervisor on board)
- ATS - Automatic Train Supervision
  management and supervision of the train operation along the whole line.

Safety Integrity Level (SIL) of Train control equipments

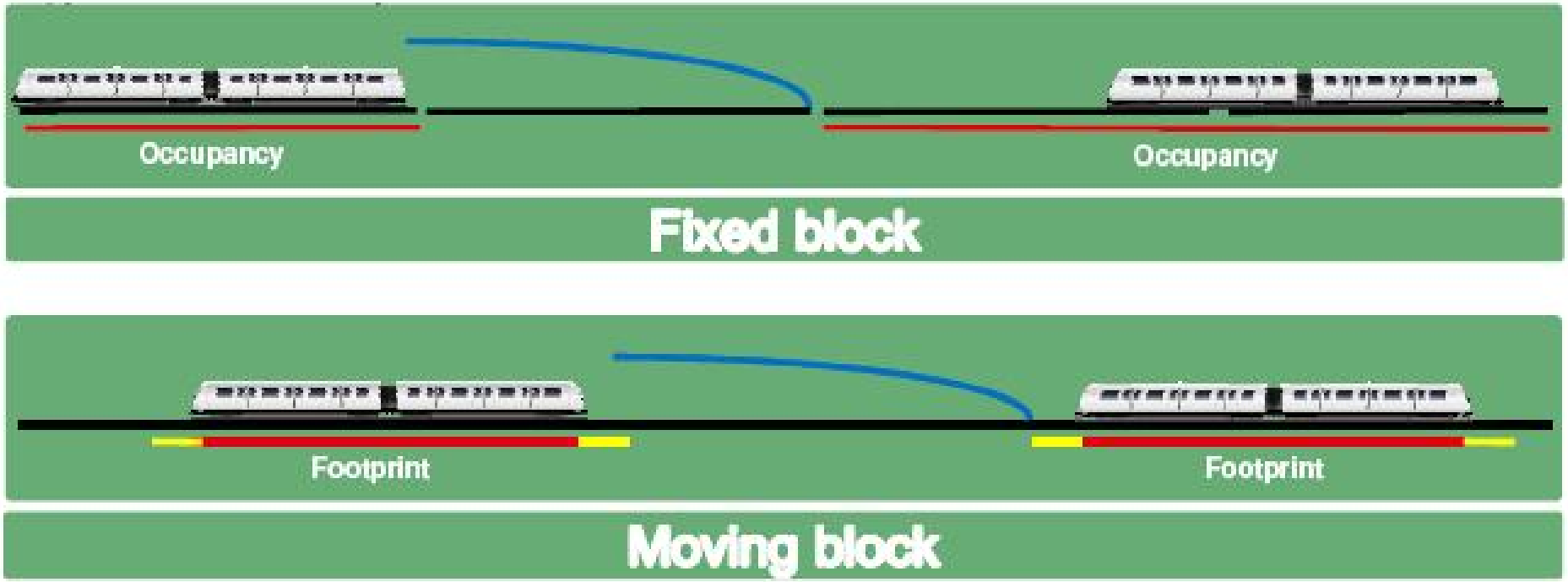
- ATP - Automatic Train Protection - **SIL4**
  The equipment protects the train from crashing into preceding train + train integrity control
- ATC - Automatic Train Control - **SIL2**
  The equipment controls whether the operation of the driver always stays in the limits dictated by the signalling and by the line topology
- ATO - Automatic Train Operation - **SIL2**
  The equipment operates the train. (UTO - Unmanned Train Operation if there is no human supervisor on board)
- ATS - Automatic Train Supervision - **SIL0**
  management and supervision of the train operation along the whole line.

ATP/ATC - Braking curve



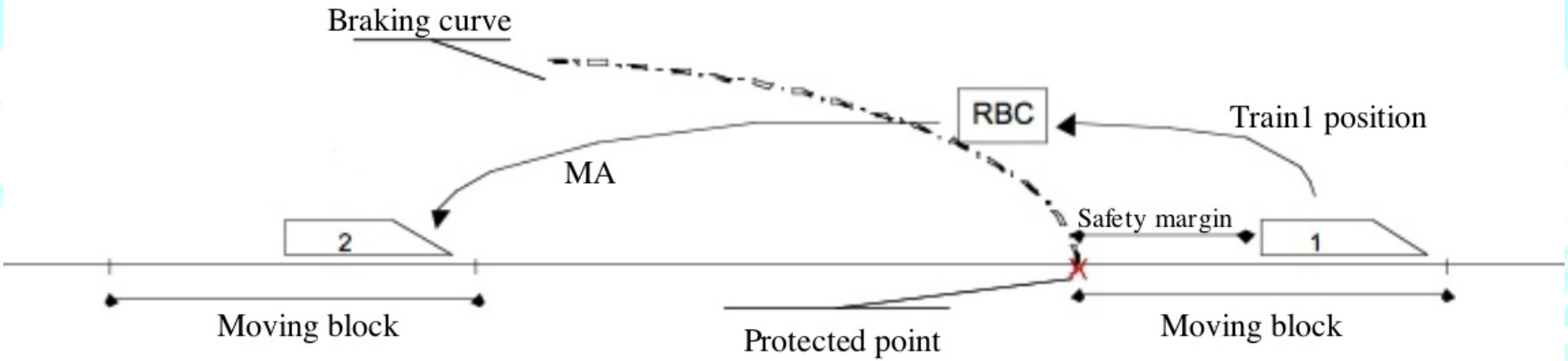
- basic concept in ATP/ATC
- safety is guaranteed if speed is always below the line.
- ATP: should the speed be above the line, emergency braking is enforced

Fixed block vs. Moving block



- *Fixed block*: line segmented into blocks, total occupancy of the leading train includes the whole block which the train is located on. Only allows the following train to move up to the last unoccupied block's border.
- *Moving block*: the train position and its braking curve is continuously calculated by the trains, and then communicated via radio to the wayside equipment, which establish protected areas, each one called Limit of Movement Authority (LMA), up to the nearest obstacle (tail of the train in front).

CBTC - Communication-Based Train Control



- RBC - Radio Block Center - communicating via GSM-R with all the trains on a section of the line
- each trains know its *exact* position (not a trivial issue), communicated to RBC
- MA - Movement Authority - continuously communicated to the following train

CBTC and Model checking

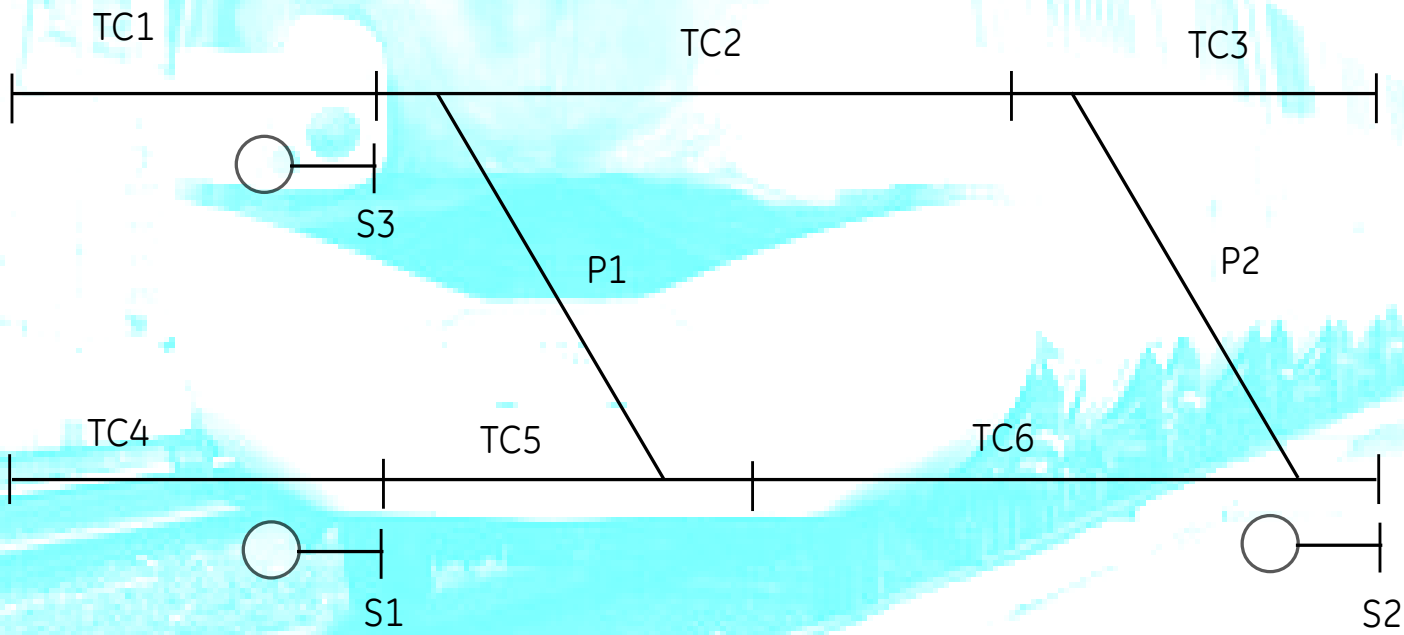
- CBTC represents the most advanced solution in ATP/ATC system
- deployed on metros (closed environments, proprietary solutions)
- to be developed inside ERTMS/ETCS level 3 (currently only level 2 is deployed)
- ISTI-CNR lab involved in a project where requirements of CBTC within ERTMS/ETCS have to be formalized
  - "classical" application of model checking: prove correctness of formalization in state diagrams according to safety properties
  - UML state diagram and UMC model checker
- possible interesting applications of probabilistic model checking:
  - probability of unsafe behaviour of the whole system
  - extraction of fault trees from the architectural specification of the system (as done in the COMPASS project)
  - modelling availability of the system

Interlocking (IXL) Systems

- An IXL system controls the movement of trains in a station and between adjacent stations
- Monitors the status of the objects in the railway yard (e.g., points, signals, track circuits)
- Allows or denies the routing of trains in accordance with the railway safety and operational regulations that are generic for the region or country where the interlocking is located

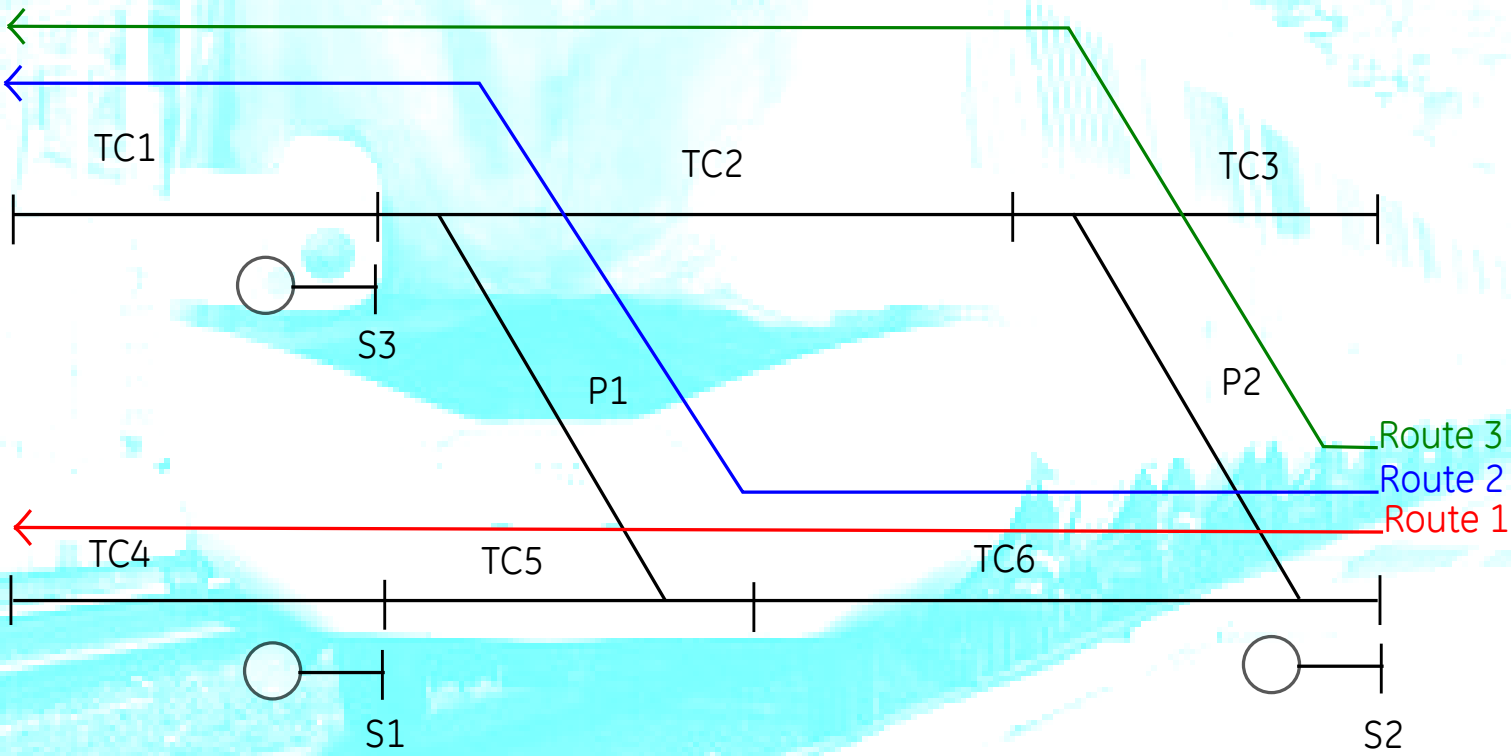
Interlocking (IXL) Systems

- An IXL system controls the movement of trains in a station and between adjacent stations
- Monitors the status of the objects in the railway yard (e.g., points, signals, track circuits)
- Allows or denies the routing of trains in accordance with the railway safety and operational regulations that are generic for the region or country where the interlocking is located



Interlocking (IXL) Systems

- An IXL system controls the movement of trains in a station and between adjacent stations
- Monitors the status of the objects in the railway yard (e.g., points, signals, track circuits)
- Allows or denies the routing of trains in accordance with the railway safety and operational regulations that are generic for the region or country where the interlocking is located

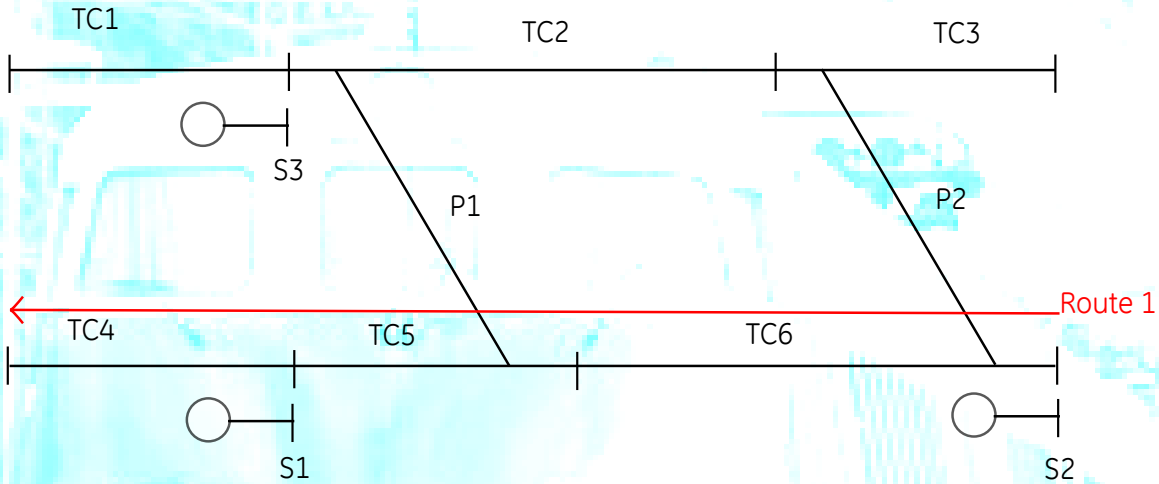


IXL Control Tables

- Instantiation of signalling rules on a station topology
- Specific for the station where the IXL system resides
- Implemented through iteratively executed software controls over the status of the yard objects

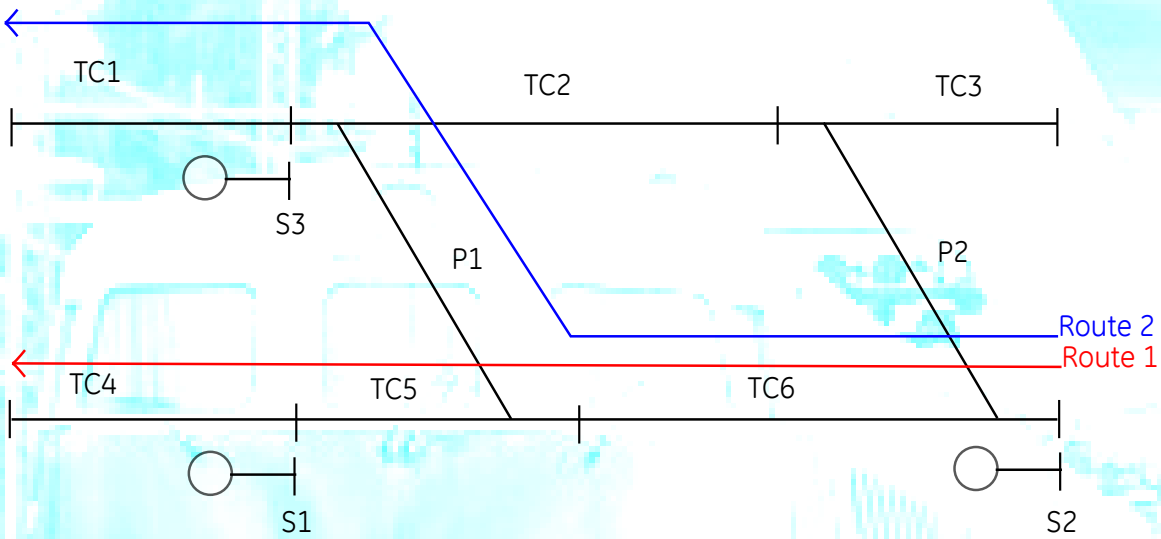
Route	S IN	S OUT	Aspect	S Ahead	Tracks	P N	P R
1	S2	S1	Y/G	R/Y	TC4 TC5 TC6	P1 P2	-
2	S2	S3	Y/G	R/Y	TC6 TC2 TC1	P2	P1
3	S2	S3	Y/G	R/Y	TC3 TC2 TC1	P1	P2

IXL Control Tables



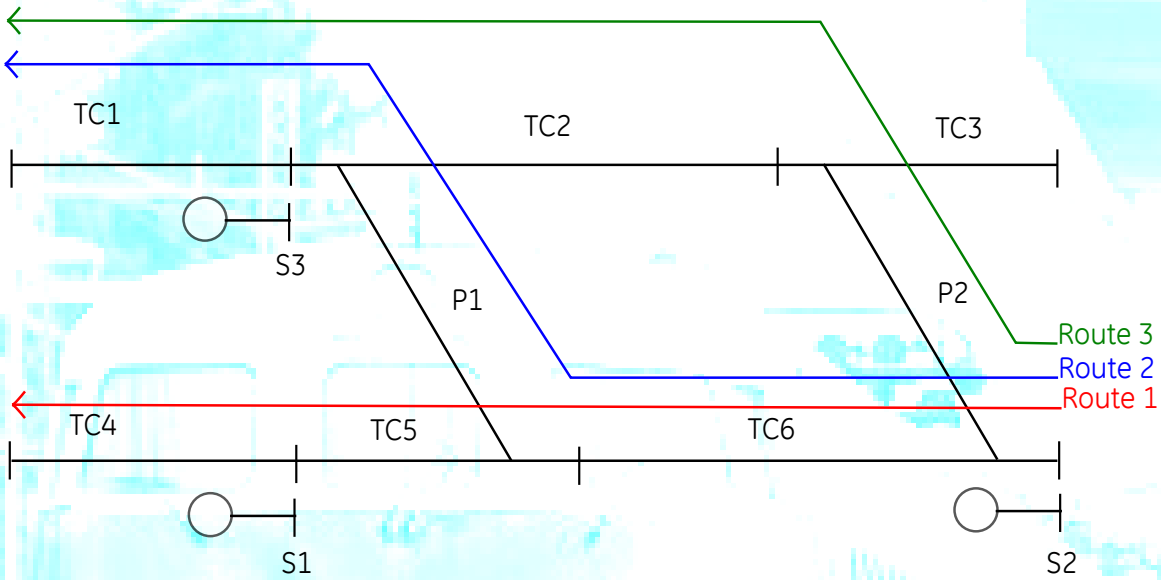
Route	S IN	S OUT	Aspect	S Ahead	Tracks	P N	P R
1	S2	S1	Y/G	R/Y	TC4 TC5 TC6	P1 P2	-

IXL Control Tables



Route	S IN	S OUT	Aspect	S Ahead	Tracks	P N	P R
1	S2	S1	Y/G	R/Y	TC4 TC5 TC6	P1 P2	-
2	S2	S3	Y/G	R/Y	TC6 TC2 TC1	P2	P1

IXL Control Tables



Route	S IN	S OUT	Aspect	S Ahead	Tracks	P N	P R
1	S2	S1	Y/G	R/Y	TC4 TC5 TC6	P1 P2	-
2	S2	S3	Y/G	R/Y	TC6 TC2 TC1	P2	P1
3	S2	S3	Y/G	R/Y	TC3 TC2 TC1	P1	P2

IXL Representation

From RIS to CIS

- Relay Interlocking Systems (RIS): logical rules implemented by means of physical relay connections
- Computer Interlocking Systems (CIS): control tables become sets of software equations executed by the IXL
- Principle schemata: relay diagrams and ladder logic diagrams
- Control table: instantiation of the principle schemata on a station topology

CIS Development Principle

As safe as the relay based equipment

IXL Representation

Disjunctive form

- Control table: set of boolean equations
- $x_i := x_j \wedge \ldots \wedge x_{j+k}$
- $x_j \ldots x_{j+k}$ are boolean variables in the form x or $\neg x$
- $x_j \ldots x_{j+k}$ represent the possible states of the signalling elements monitored by the control table: system input, output or temporary variables
- Equations are conditional checks over the current and expected status of the controlled elements

Problem dimension

- $size = (m, n)$
- m is the maximum number of inter-dependent equations involved
- n is the number of inputs of the control table
- independent set of equations can be verified separately

IXL Execution Model

- the equations are "interpreted" by a reasoner engine.
- The reasoner engine is the same for every plant
- The logic is coded as "data", for the reasoner
- Behind this choice is the criterion of minimization of certification efforts: the reasoner is certified at SIL4 once for all,
- The data should be validated for each plant, but are "easier" to certify if they can be related in some way to the traditional "principle schemata" adopted by railway engineers in the era of relay-based IXLs

IXL Safety Requirements

- Assess correctness of a control table design
- Correctness is expressed with respect to safety properties

Modelling the interpretation of boolean equations in NuSMV and SPIN

- Investigate the applicability upper bound on the size of IXL problems that can be handled by general-purpose model checkers (Ferrari, Magnani, Grasso, Fantechi, FORMS-FORMAT 2010)
- Random generation of "reasonable" boolean equations sets of increasing size

Safety properties representation

- Haxthausen, A. E., & Peleska, J. (1999)
- CTL: AGAX form: $AG(p \rightarrow AXq)$
- LTL: GX form: $G(p \rightarrow Xq)$

IXL Safety Requirements

- $AG(p \rightarrow AXq)$ are fail-safe conditions
- Events that will happen if some unsafe condition occurs
- Property that shall hold when certain controls are active

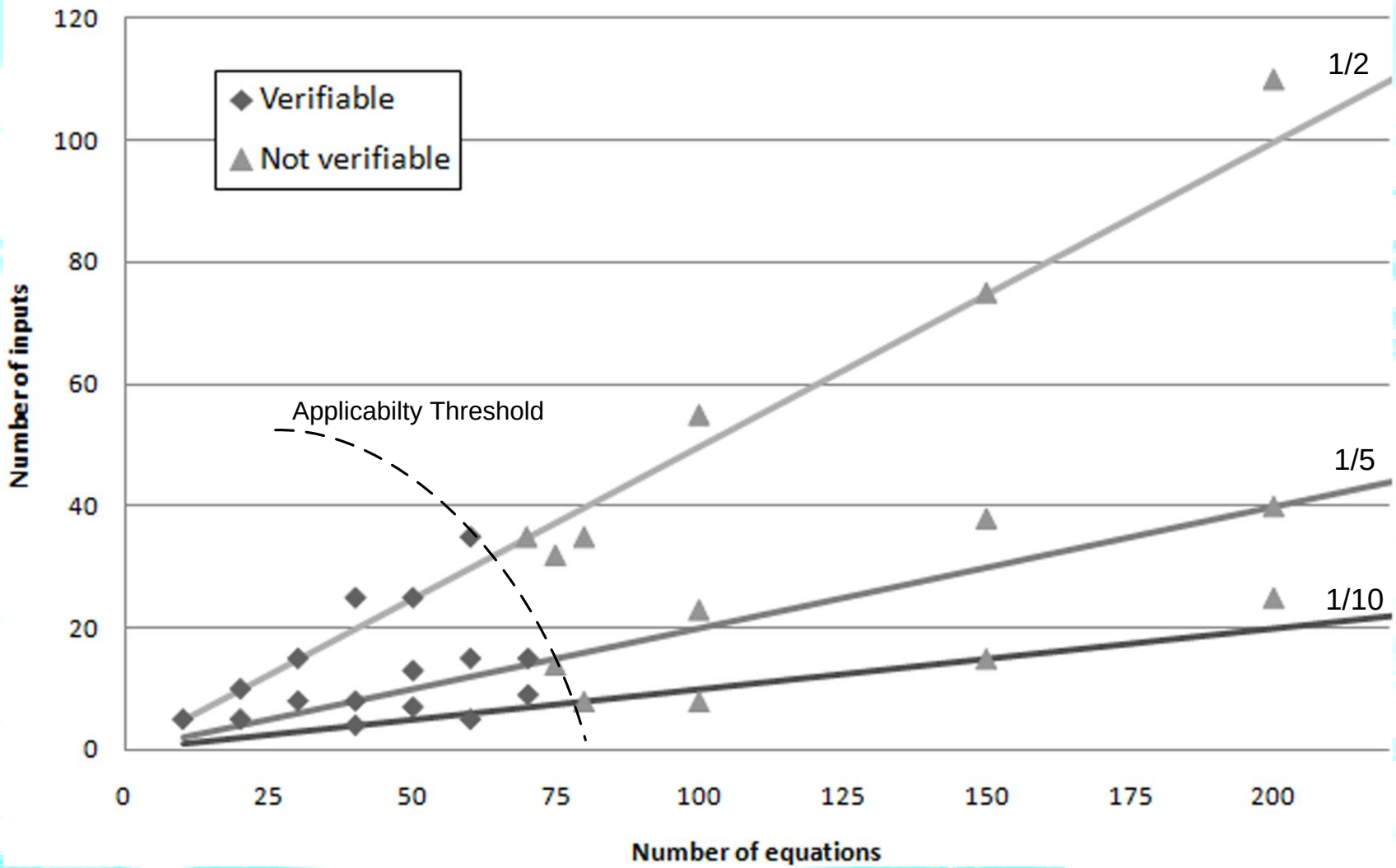
Example property - No Derailing

While a train is crossing a point, the point shall not change its position

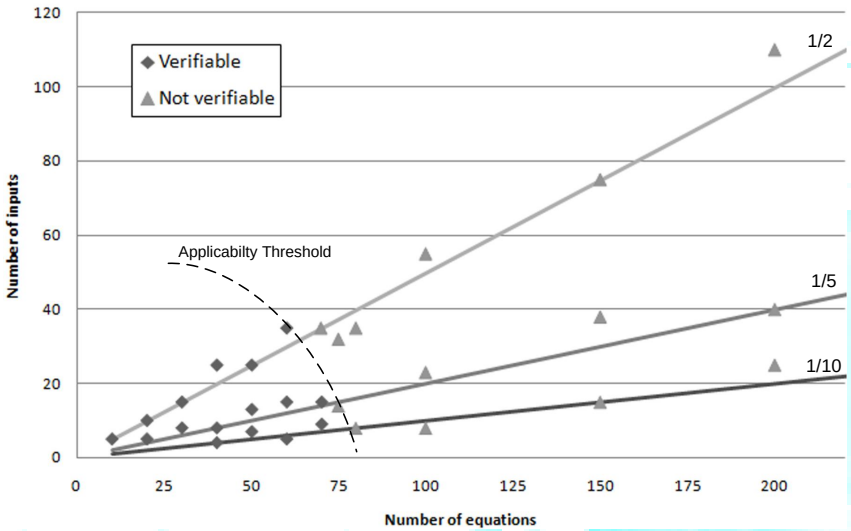
- $AG(occupied(tc_i) \wedge setting(p_i) = val \rightarrow AX(setting(p_i) = val))$
- Whenever the track circuit tc_i associated to a point p_i is occupied, and the point has the proper setting *val*, this setting shall remain the same on the next state

Generation of test boolean equations sets that "by construction" satisfy properties expressed in CTL-AGAX form, in order to force the worst-case full state space exploration

NuSMV Results

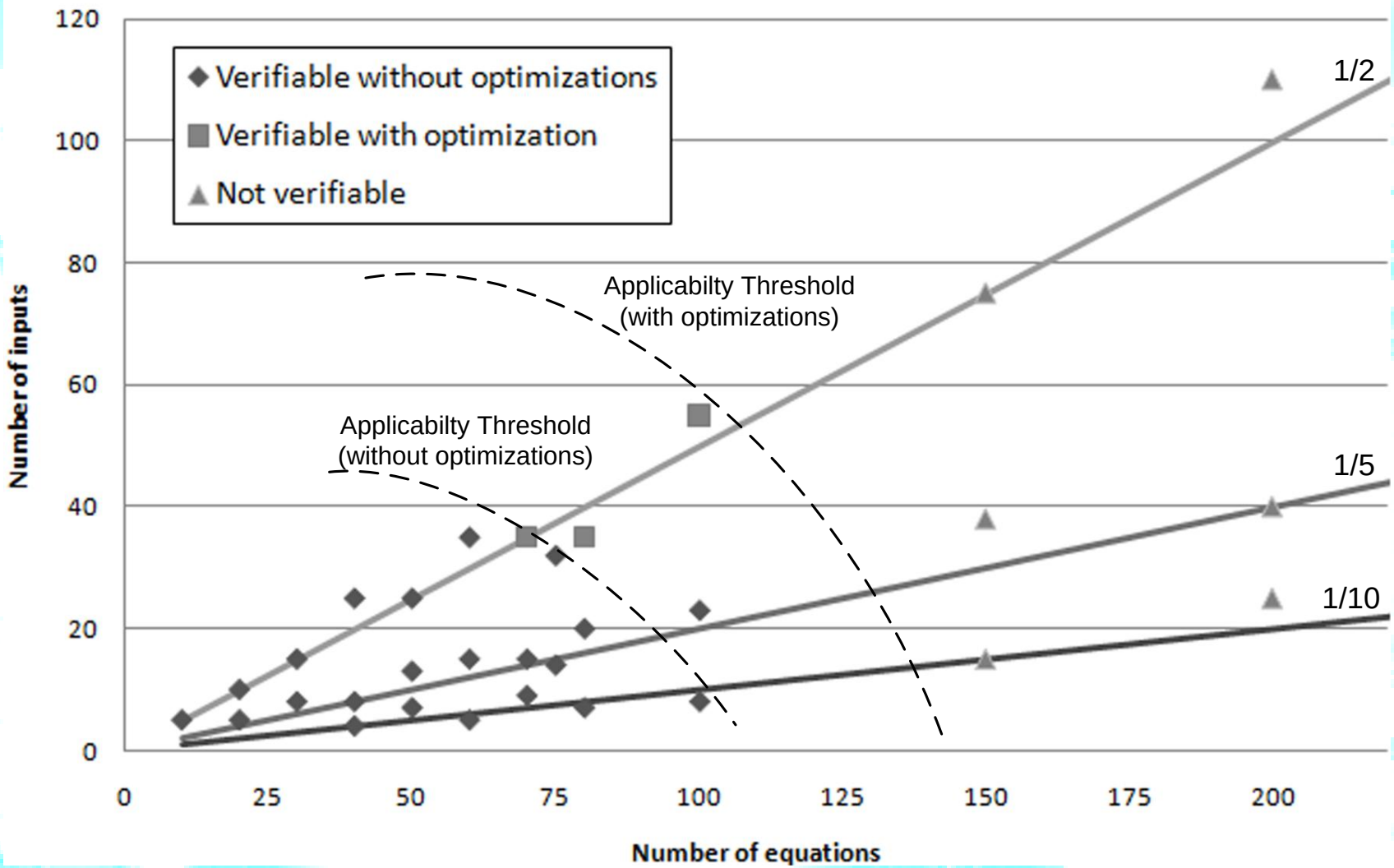


NuSMV Results

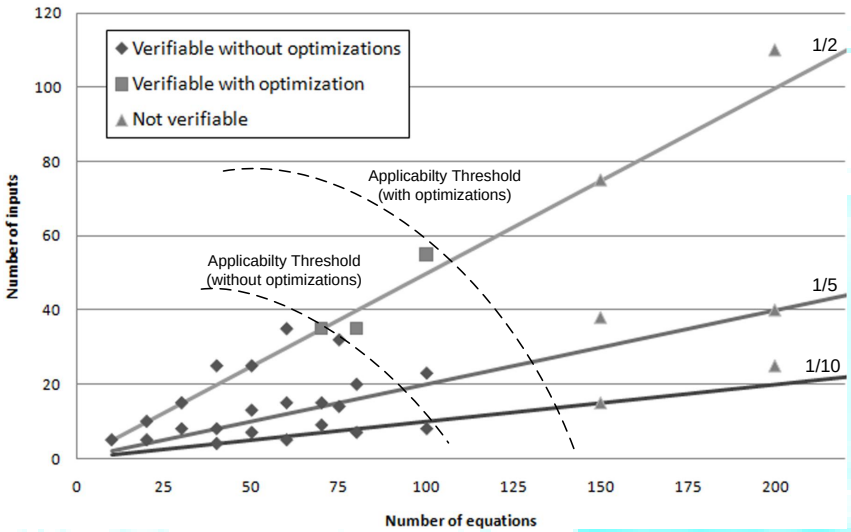


- Non-verifiable cases
  - memory exhaustion (4Gb)
  - 36 hours of execution without answer
- Ratio 1/10 to 1/5: 70 equations
- Ratio 1/2: 60-65 equations
- No improvements with optimization

SPIN Results



SPIN Results



- Non-verifiable cases
  - memory exhaustion (4Gb)
  - increasing input/equation ratio causes crash
- 80 equations and 20 inputs without optimization
- Minimized Automata optimization: minimal deterministic state automata
- 100 equations and 60 inputs

Discussion

Model checker comparison

- The two model checkers show comparable results without optimization
- SPIN is more influenced by the number of input variables: increasing input causes state space explosion
- NuSMV optimizations do not increase performances
- Slightly better results with SPIN using Minimized Automata

Model checking applied to interlocking systems of medium size (normally some hundreds of equations) is unfeasible

Discussion of the results

Applicability limits

- Worst-case analysis
- Results are given for true safety properties: properties issuing counterexamples are expected to give better results
- Verification harder than falsification: Model checking used for certification is more expensive than model checking used for "bug hunting".
- Results are given on sets of inter-dependent equations
- Slicing w.r.t. interdependency can be applied only if the actual topology of the tracks layout and the interlocking functionality do separate concerns about different areas of the layout
- In a real world interlocking, we could attempt to verify slices of about 7 track circuits
- In real world interlockings slices are normally larger

Related Work

K. Winter, University of Queensland.

- BDD-based NuSMV.
- improving efficiency by by customising the ordering of state variables occurring in the model to be checked.
- In the domain of railway interlockings represented as control tables this task can be supported using an algorithm that has access to the track layout information.
- With such optimization strategies symbolic model checking is feasible for quite large scale interlocking systems. (e.g., a system with 41 routes, 9 points, 19 signals, and 31 track circuits is not feasible with default variable ordering, but can be addressed with the optimization algorithm)

SAT-based model-checking

- The NuSMV SAT-based solver behaves much better than that of the BDD-based one, so that a size of 200 equations with 20 inputs can be addressed. However the SAT-based solver is used in a Bounded Model Checking scheme which cannot prove safety properties unless the chosen depth is enough to cover all the cycles of the model.
- A few experiments conducted within General Electric Transportation Systems on Stateflow models of real medium scale interlockings with Mathworks' Design Verifier (based on a SAT engine by Prover Technology) show it as able to deal with such models, although apparently at the limits of its capacity. Actual figures are not available.
- Indeed Prover Technology has launched a commercial solution (Ilock) for the production of interlocking software, that includes formal proof, by means of a SAT solving engine, of safety conditions. Bounds on the addressable size of the controlled yard and strategies used to address large state spaces are not known.
- an optimized problem-tailored boolean encoding of the equations and of the execution model should be investigated for application of an efficient SAT-solver,

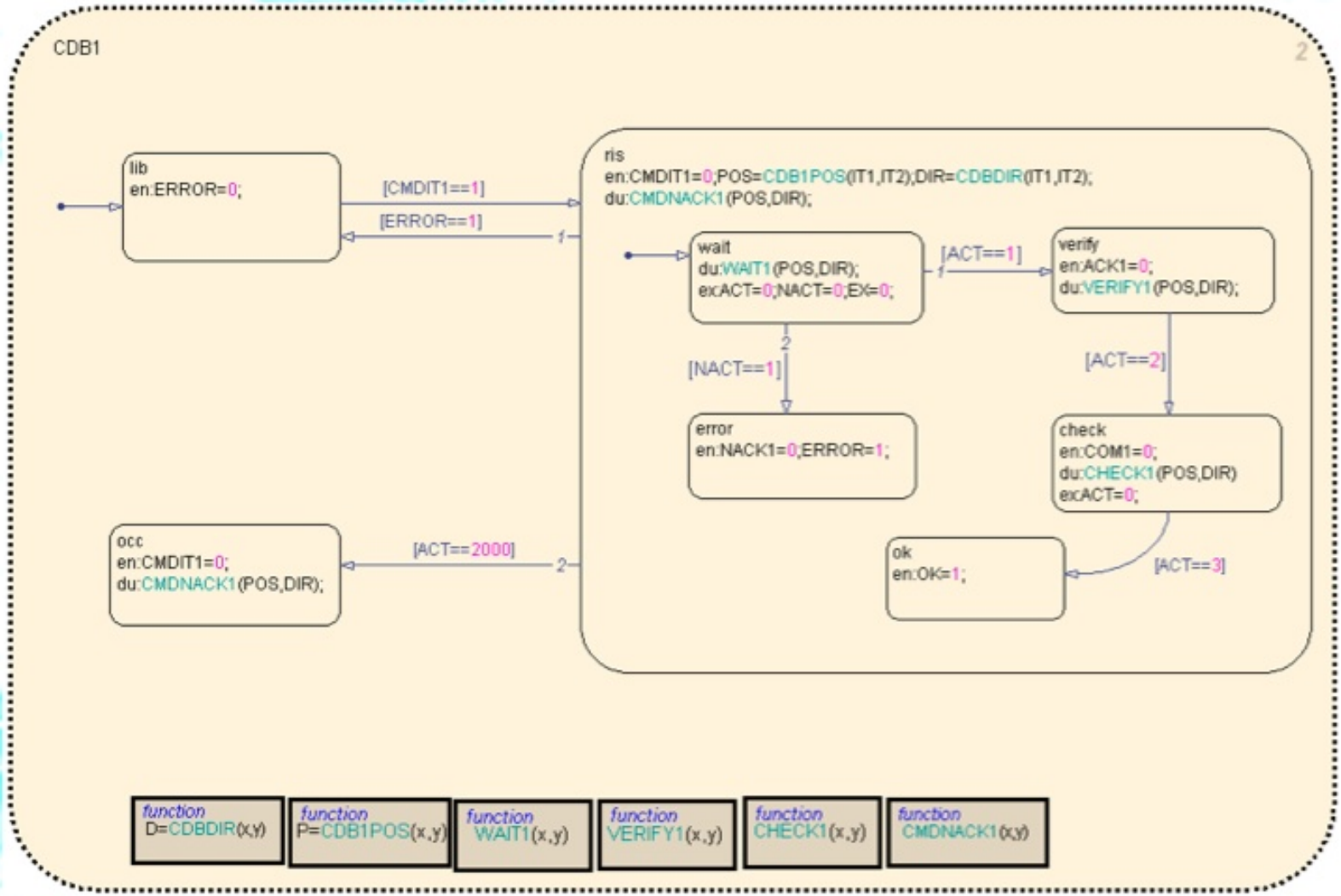
Distributed IXL

- The elements of the geographic approach can be configured as a set of distributed, communicating, processes
- Each element controls a given layout element
- A global notion is instead that of route
- A route has to be estabilshed by proper cooperation of the distributed elements
- The communication among processes follows the physical layout of the station/yard
- Hence, a route is established by the status of the elements that lie along the physical route

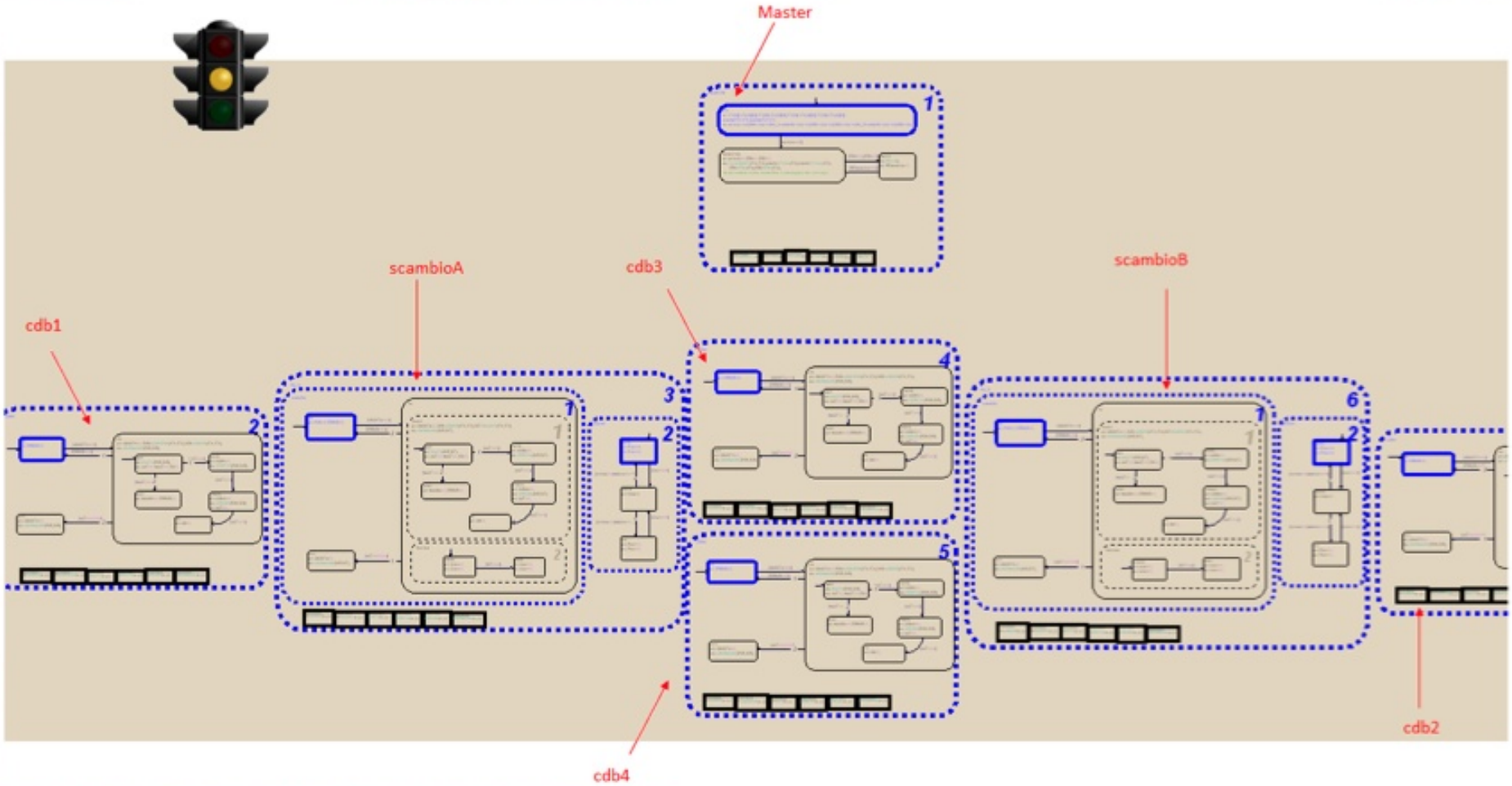
Distributed IXL

- Experiments done, at the distributed logic level, by modelling in {SCADE, Stateflow,UML} of a distributed IXL:
  - Tracks and points modeled as statecharts
  - Communications between adjacent objects
  - Route establishment protocol based on the classical two-phase commit protocol
- The two phase commit protocol guarantees that the route is established only if all the elements on the route are reserved and locked
- Formal verification via Model checking of basic IXL safety proeprties (*e.g. no-derailment property*)

Stateflow track circuit model



Replication and distribution



From CIS to DIS

- Instantiation of DIS from Control Tables (Banci, Fantechi, Gnesi - FMICS 2005), basing on work done at RFF-SNCF.
- More recently, the formalization of interlocking rules carried on in the INESS project follows the same direction.
- Preliminary results indicate that even for quite large IXLs, states spaces appear to be affordable, provided few trains are modeled, due to the "locking" properties of the system. Limiting trains to two limits concurrent behaviour.
- Proving no-collision properties with only two trains is not a limitation, since from a physical point of view interactions between more than two trains are "incredible" scenarios
- if these results are confirmed by further experimentation, it would be a point in favour of DIS w.r.t. boolean equation-based CIS, which is hardly abandoned by most railway infrastructure companies

Interlocking as a Product Family

- Manufacturers of (boolean-equations based) interlockings in practice adopt a *product-line* approach - without knowing it:
  - *commonalities*: the reasoner, invariable for any deployed interlocking
  - *variabilities*: the station layout, coded first in Control Tables and then in boolean equations.
  - Any single deployed interlocking is actually a product of the family
  - Certification is factorized among products, for what concerns the reasoner
- In a DIS it is more difficult to elicit a product-line:
  - *commonalities*: models of each object as a single process, class definitions of each object: let's call them *features* in accordance to product line terminology.
  - *variabilities*: multiplicity of objects and their connections.
  - Configuration of a single plant by instantiation of generic elements acording to layout and Control Tables
  - Can we factorize part of the certification effort? Studies on inferring model checking results form families to products are at their dawning - the gap to extend such results to this aim is very large

Conclusion & Future Work

- really few conclusions....
- much future work....
-
-
- Thanks!