This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643921.

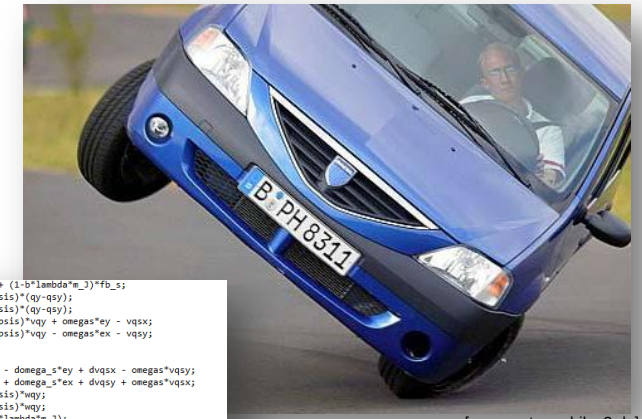
Automated Driving and Conformance Testing

Daniel Heß, DLR

Alexander Rausch, Bosch

UnCoVerCPS Workshop – 2018/6/6

CHALLENGES OF CPS

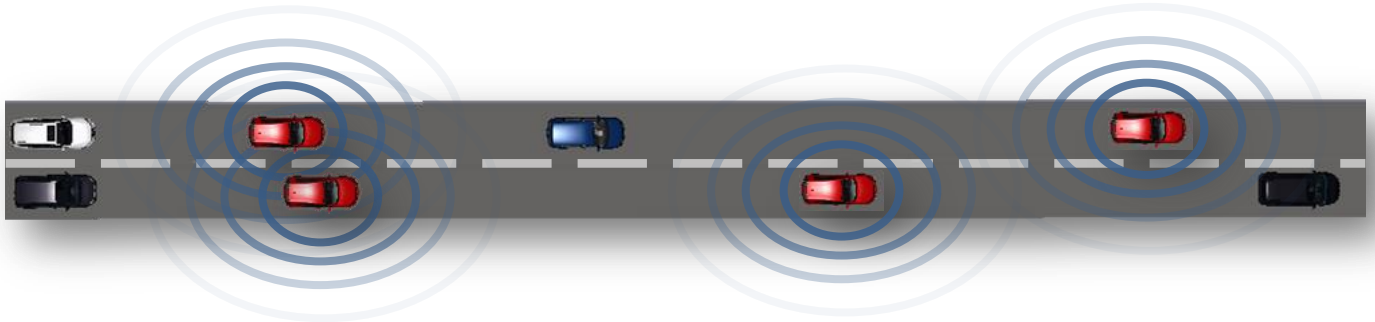


[www.motormobiles2.de]

```
double dvqsy = -omegas*vqsx + (1+a*lambda*m_J)*fa_s + (1-b*lambda*m_J)*fb_s;
double   ex = std::cos(-psis)*(qx-qsx) - std::sin(-psis)*(qy-qsy);
double   ey = std::sin(-psis)*(qx-qsx) + std::cos(-psis)*(qy-qsy);
double  dex = std::cos(psi-psis)*vqx - std::sin(psi-psis)*vqy + omegas*ey - vqsx;
double  dey = std::sin(psi-psis)*vqx + std::cos(psi-psis)*vqy - omegas*ex - vqsy;
double ddex = -kIx * Iex - k0x * ex - k1x * dex;
double ddey = -kIy * Iey - k0y * ey - k1y * dey;
double wqx = ddex - omegas*omegas*dex - 2*omegas*dey - domega_s*ey + dvqsx - omegas*vqsy;
double wqy = ddey - omegas*omegas*dey + 2*omegas*dex + domega_s*ex + dvqsy + omegas*vqsx;
double aqx = std::cos(psi-psis)*wqx - std::sin(psi-psis)*wqy;
double aqy = std::sin(psi-psis)*wqx + std::cos(psi-psis)*wqy;
double fay_required = (aqy-(1-b*lambda*m_J)*fb)/(1+a*lambda*m_J);
double fx_required = (aqx + omega*omega*lambda);
double fax_required = fx_required;
if(fax_required>0.0)//positive acceleration by engine
{
    fax_required *= abf;//assign front ratio for driving force
}
else//negative acceleration by brake
{
    fax_required *= bbf;//assign front ratio for braking force
}
double fa_max = mu_ctrl*mu*g*b/(a+b);
double force_ratio = std::sqrt(fax_required*fax_required + fay_required*fay_required)/fa_max;
if(force_ratio>1.0)//force exceeds admissible maximum ==> scale down the requested force
{
    fx_required /= force_ratio;
    fay_required /= force_ratio;
}
```

- Mixed integer and continuous states and transitions
- Non-determinisms (unreliable sensors, actuators and models)

CHALLENGES OF CPS



- Heterogeneous, distributed system
- Unreliable communication

CHALLENGES OF CPS

[Kölner Stadtanzeiger]

- Huge numbers of environmental factors

CHALLENGES OF CPS

[news.vice.com]

- Huge variety of situations
- Unexpected and rare situations

CHALLENGES OF CPS

[techrepublic.com]

[www.quora.com]

- Complex and time-variant controls

CHALLENGES OF CPS

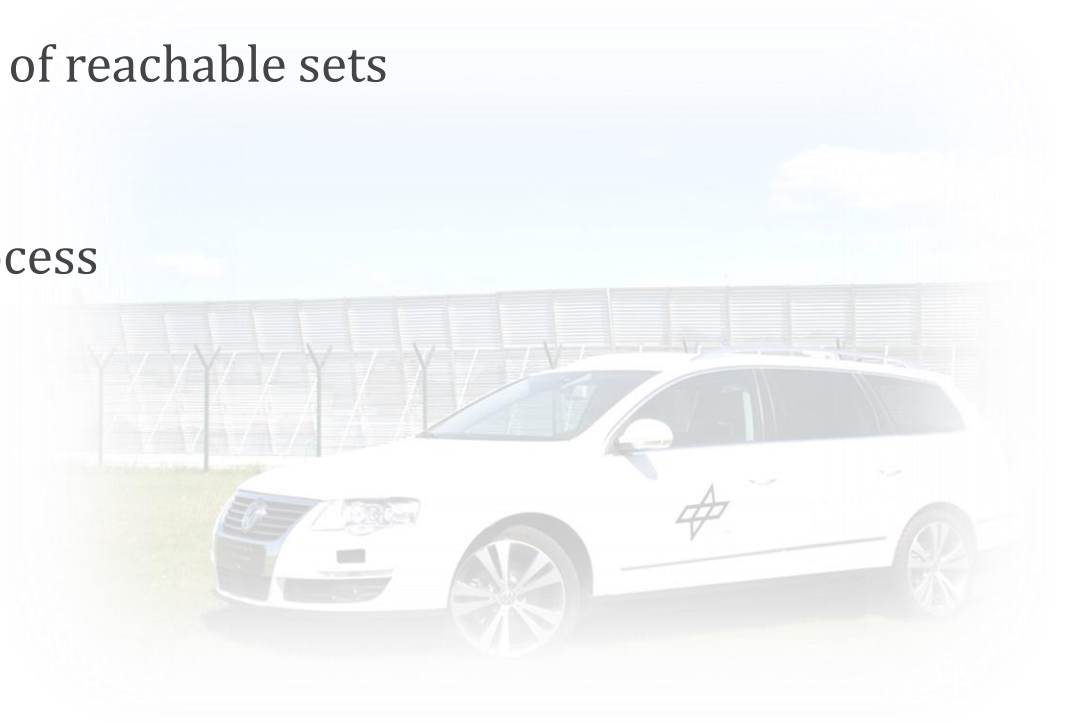
- **Offline validation** of CPS is difficult
  - Size of offline verification problems becomes un-manageable
  - Enormous number of tests to achieve required coverage

N. Kalra and S. M. Paddock, 2016: *440 million km test drive, to show with 95% reliability that an automated vehicle causes less accidents than an average human driver*

- **Online Verification** in EU project UnCoVerCPS
  - Verify safety of an action during operation of the system
  - Account for uncertainties with worst-case assumptions
  - Investigation of the approach on the example of automated vehicles

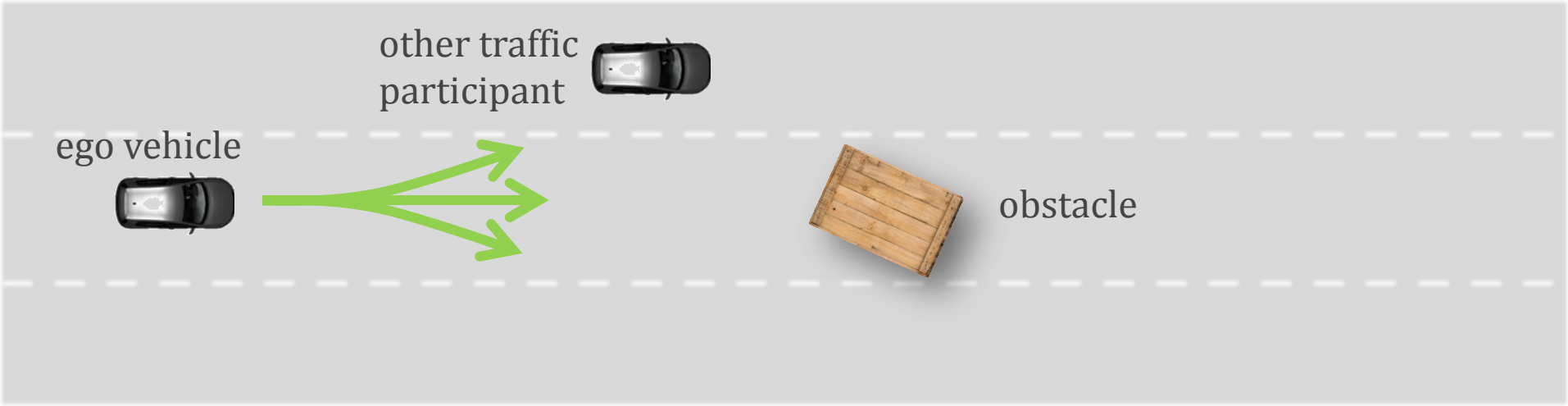
AGENDA

- Invariant Safety
- Approach
  - Reachability Analysis for Ego Vehicle
  - Offline Pre-computation of reachable sets
  - Online Verification
- Architecture and Design Process
- Conformance Testing
- Results
- Discussion



INVARIANT SAFETY

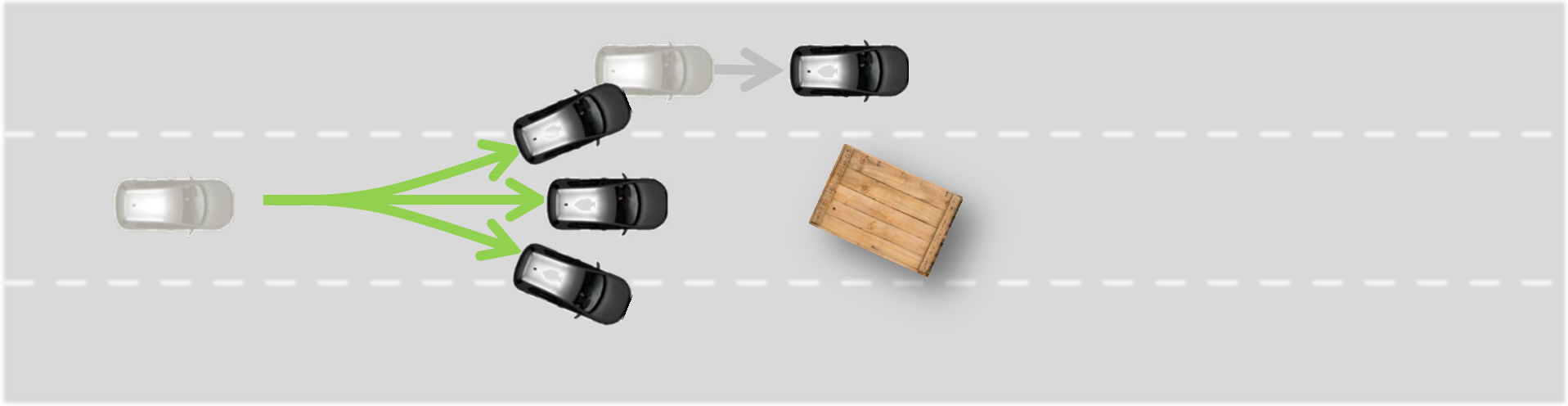
Which action is safe?



INVARIANT SAFETY

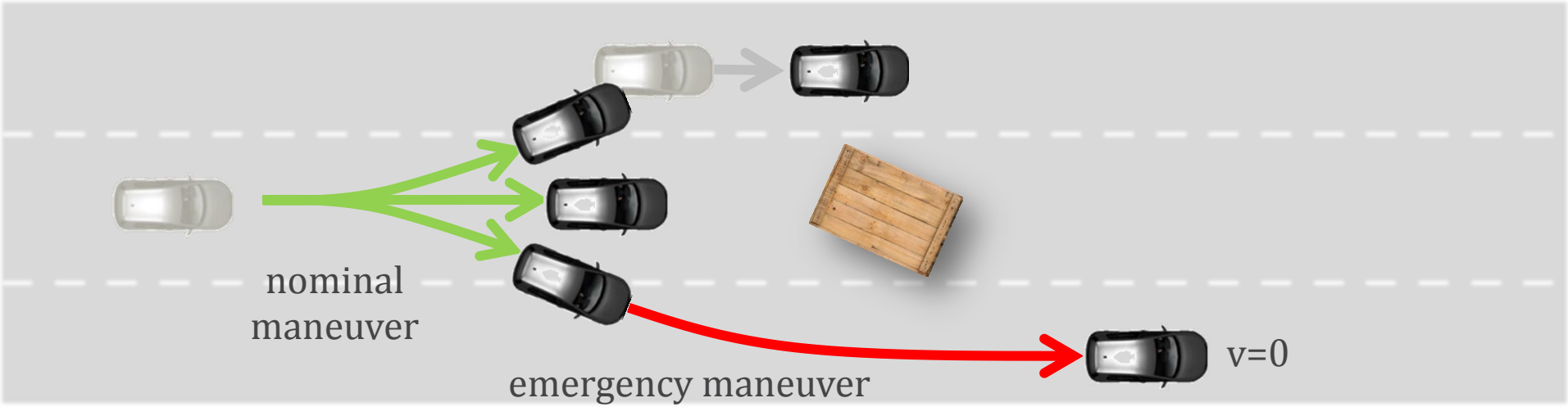
Which action is safe?

Always depends on the next actions, until standstill.



INVARIANT SAFETY

Define standstill in a certain lane to be a safe state



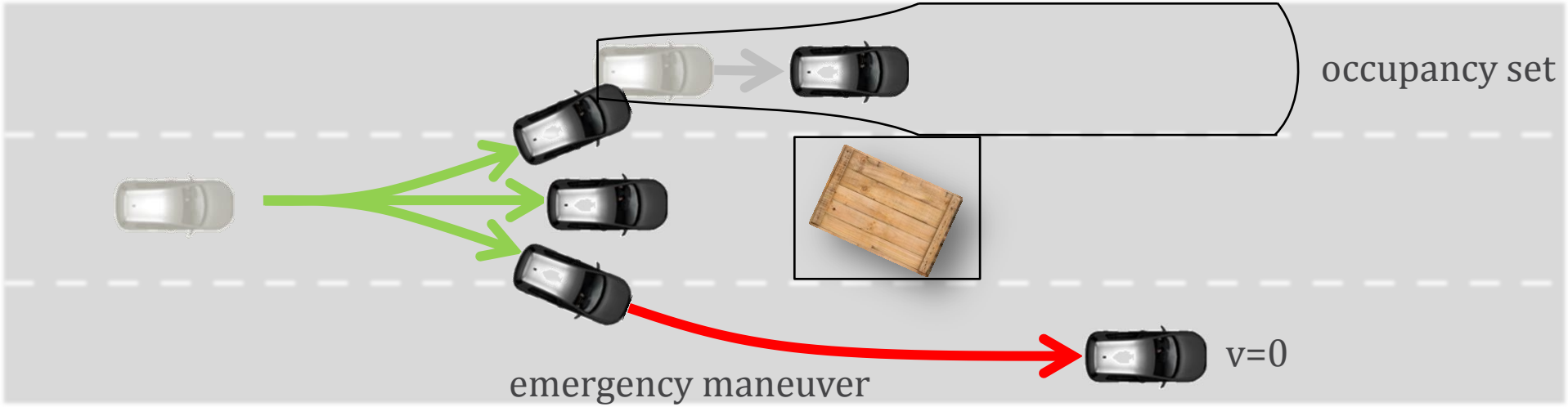
Show that safe state can be reached

- after execution of nominal action
- under all legal behaviors of others
- under uncertainty of ego

⇒ Action is safe!

INVARIANT SAFETY

Define standstill in a certain lane to be a safe state



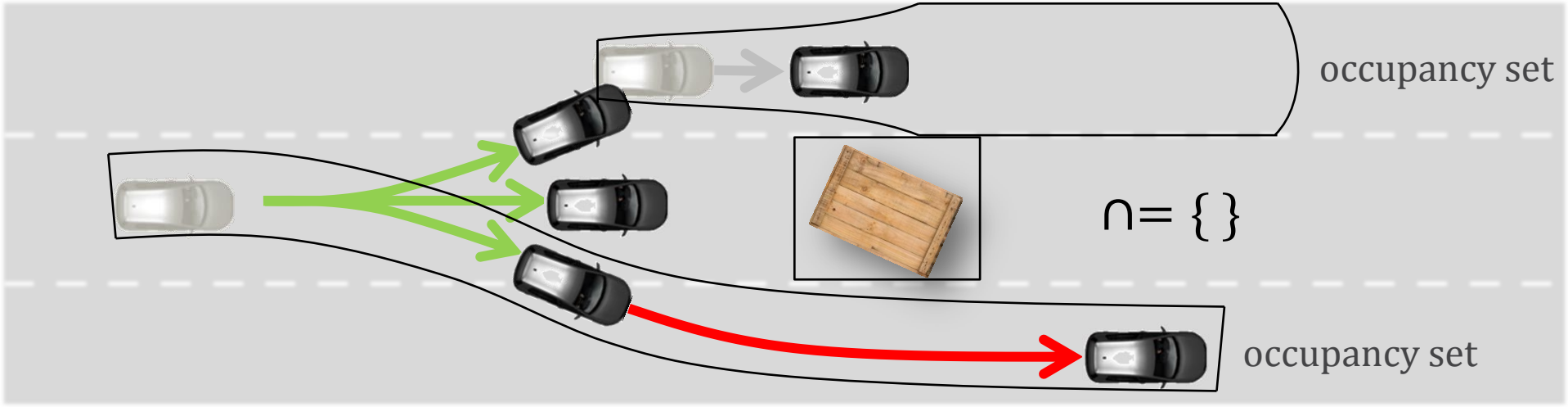
Show that safe state can be reached

- after execution of nominal action
- under all legal behaviors of others
- under uncertainty of ego

⇒ Action is safe!

INVARIANT SAFETY

Define standstill in a certain lane to be a safe state



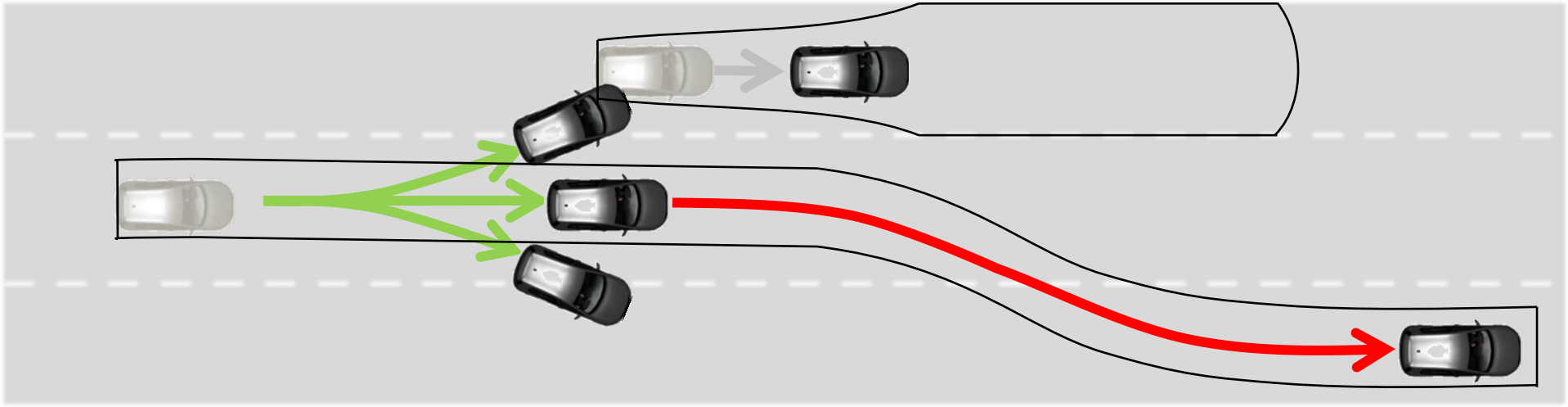
Show that safe state can be reached

- after execution of nominal action
- under all legal behaviors of others
- under uncertainty of ego

⇒ Action is safe!

INVARIANT SAFETY

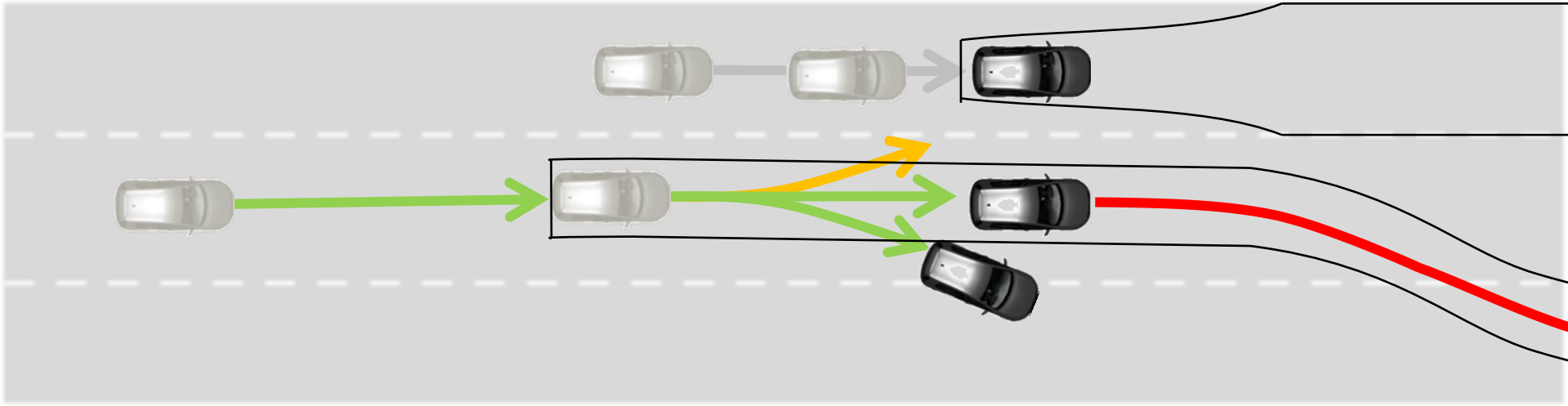
Repeat proof of invariant safety



- If a new emergency maneuver can be found for new nominal maneuver
  - Execute new nominal maneuver
- Otherwise: Execute old emergency maneuver

INVARIANT SAFETY

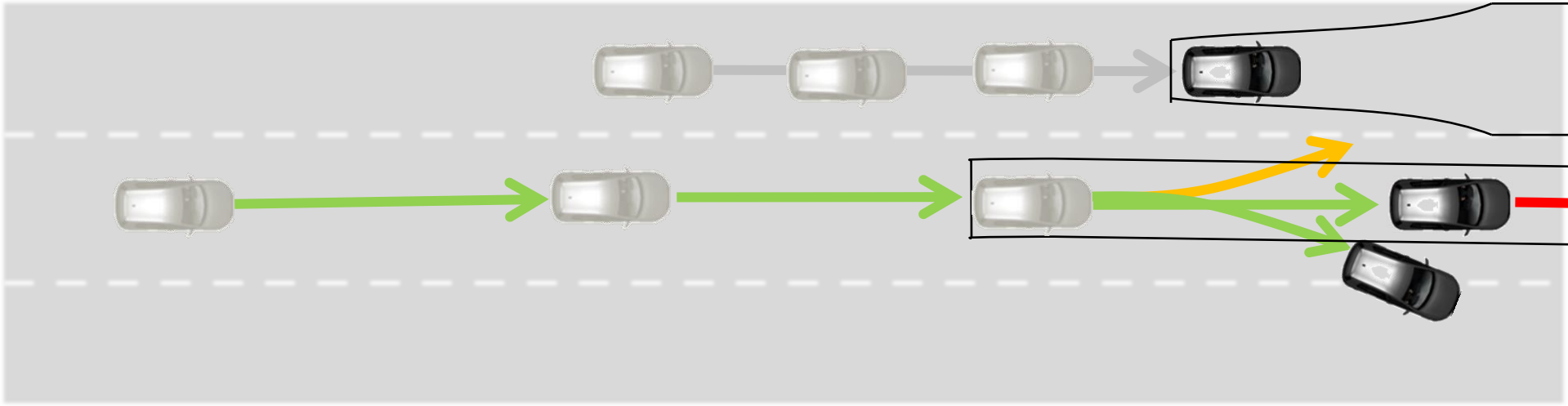
Repeat proof of invariant safety



- If a new emergency maneuver can be found for new nominal maneuver
  - Execute new nominal maneuver
- Otherwise: Execute old emergency maneuver

INVARIANT SAFETY

Repeat proof of invariant safety



⇒ Keep driving safe, nominal maneuvers while available in given traffic situation

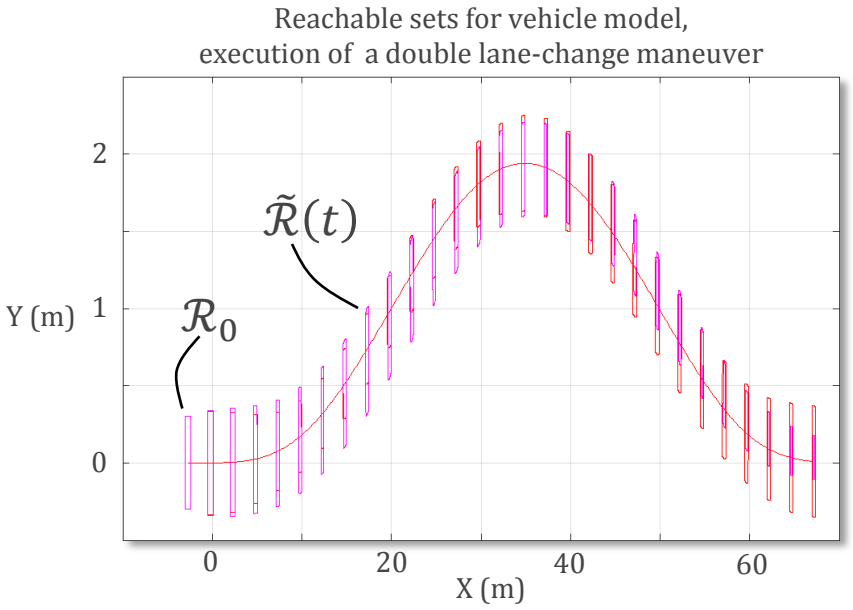
OCCUPANCY OF THE EGO VEHICLE

- Reachability analysis for dynamical systems
  - $\dot{x} = f(x, u, e), \quad x \in \mathbb{R}^n, \; u \in \mathbb{R}^m, \quad e \in E \subset \mathbb{R}^k$ bounded error
  - Stabilize: $\dot{x} = f(x, c(x, x^*), e) =: f_c(x, e)$
  - Flow: $\dot{\Phi}(x_0, e(\cdot), t) = f_c\big(\Phi(x_0, e(\cdot), t), e(t)\big); \quad \Phi(x_0, e(\cdot), 0) = x_0$
  - Reachable set $\mathcal{R}(t) = \{\Phi(x_0, e(\cdot), t) \mid e(\cdot) \in E, x_0 \in \mathcal{R}_0\}$

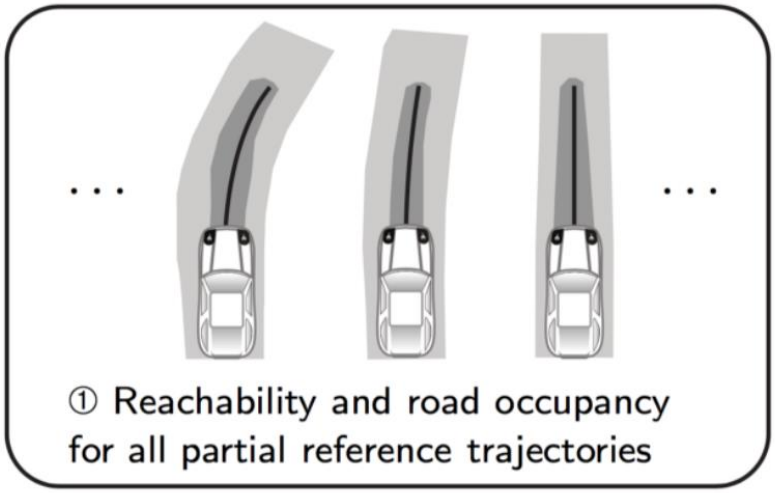
- CORA computes $\tilde{\mathcal{R}}(t) \supseteq \mathcal{R}(t)$

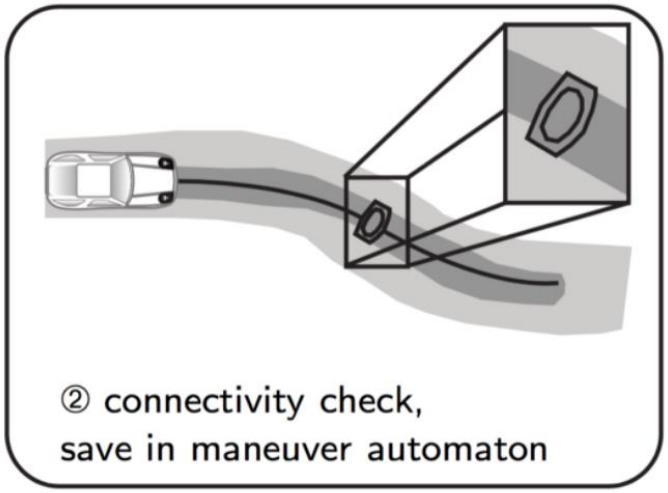
  $\Rightarrow$ Guaranteed over-approximation

- Offline pre-computation of reachable sets for real-time maneuver planning



OFFLINE PRE-COMPUTATION





- Vehicle model f
- Controller c
- Validated error bounds E

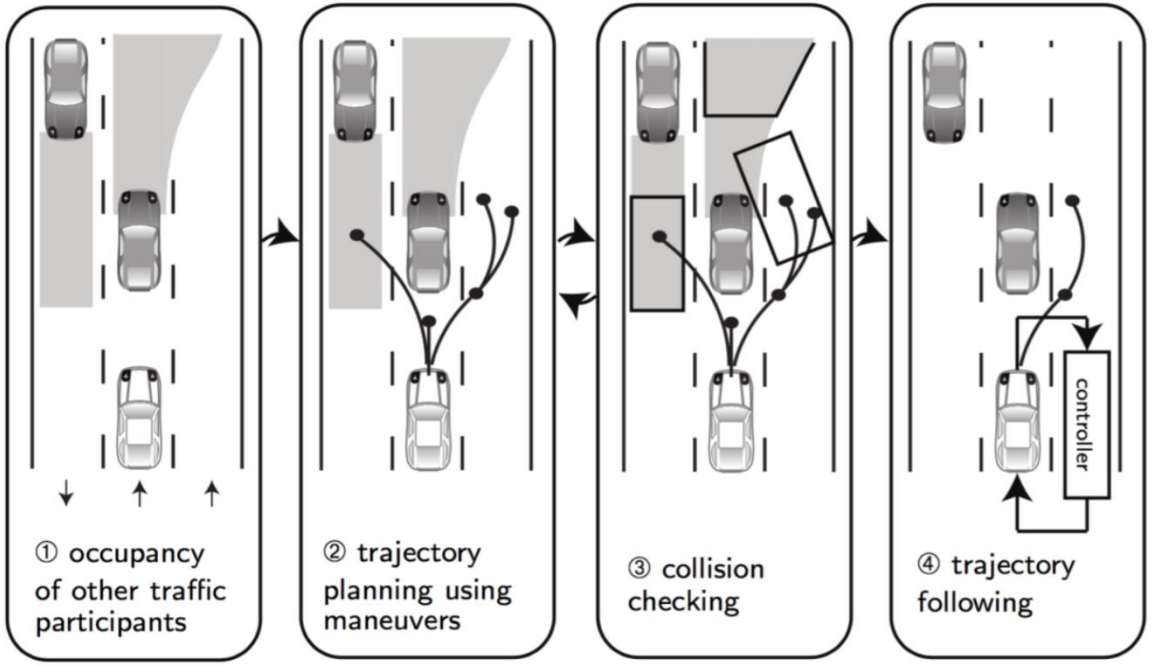
In: Non-deterministic, closed loop model f_c,

⇓

Out: Safe maneuver automaton

- Reference trajectories
- Resulting reachable sets
- Occupancy
- Safe transitions between maneuvers

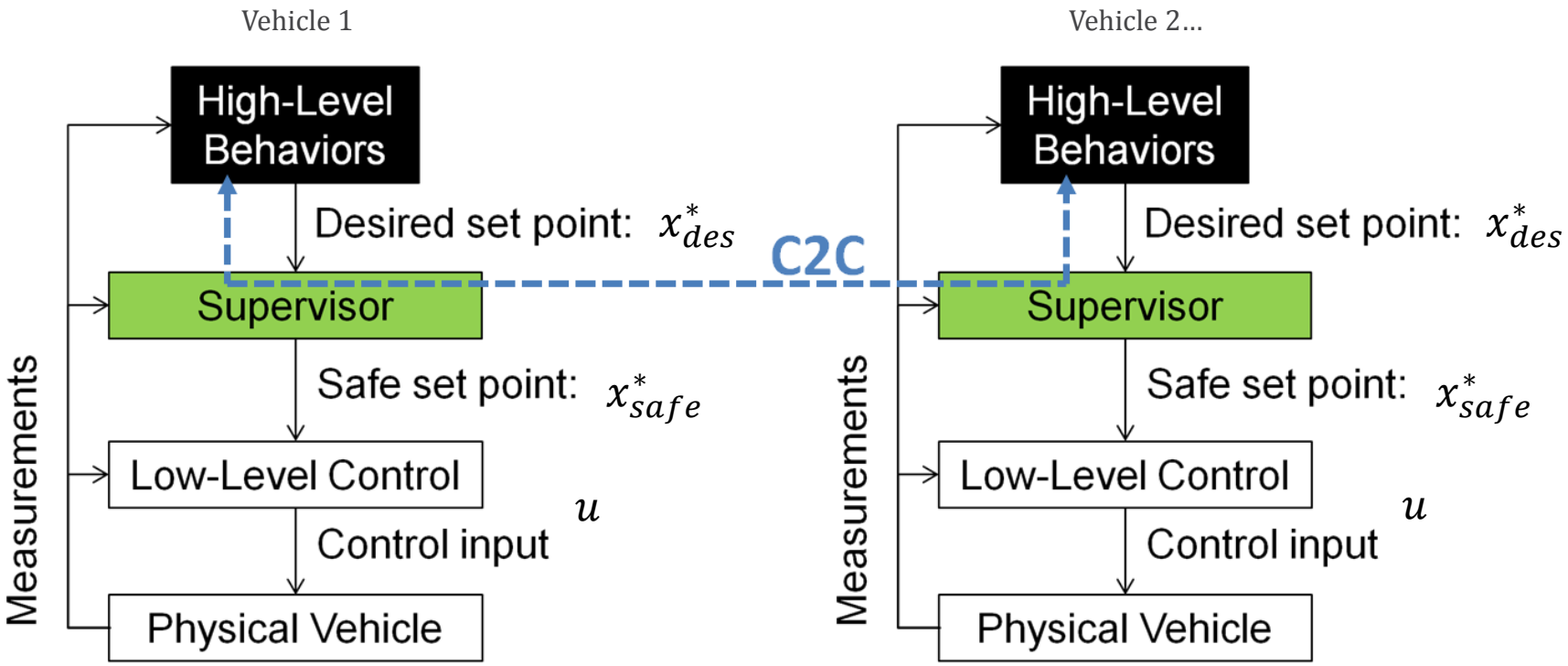
ONLINE VERIFICATION



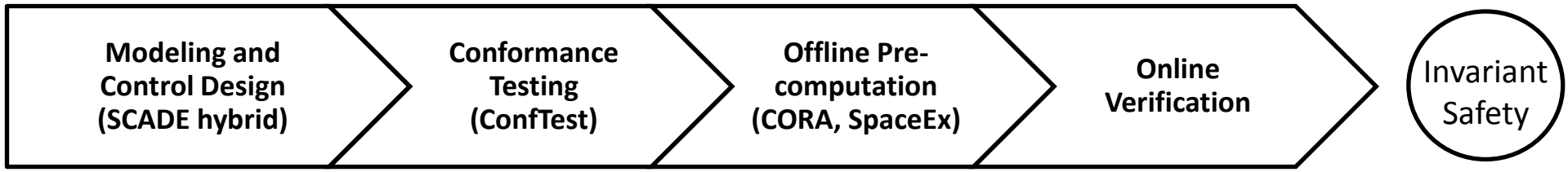
- **In**: Ego state, scene, desired nominal trajectory
- **Out**: Existence of safe emergency maneuver

ARCHITECTURE

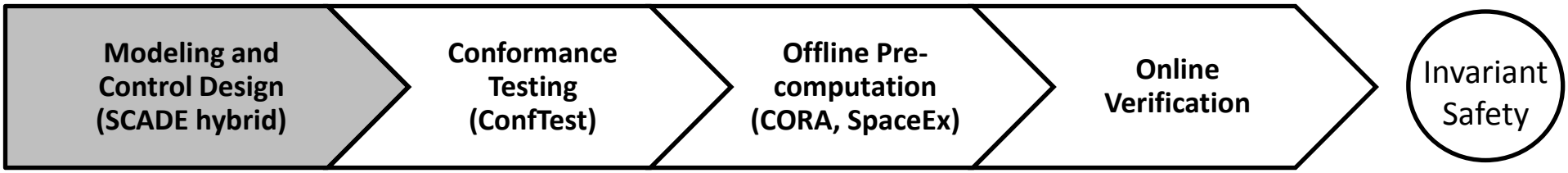
- Proposed architecture for safe, cooperative, automated driving
- Assumes wireless car-to-car (C2C) communication



DESIGN PROCESS

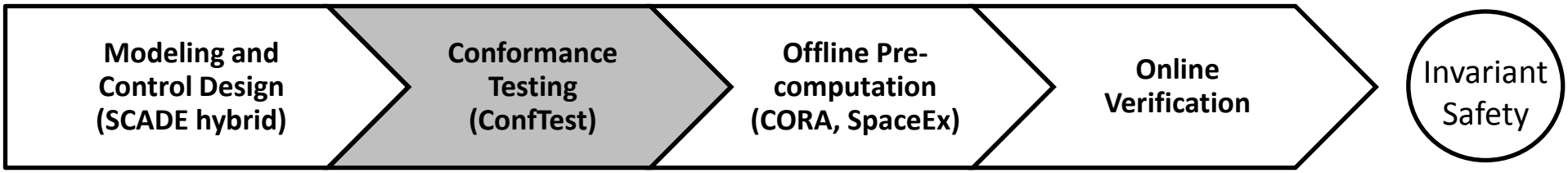


DESIGN PROCESS



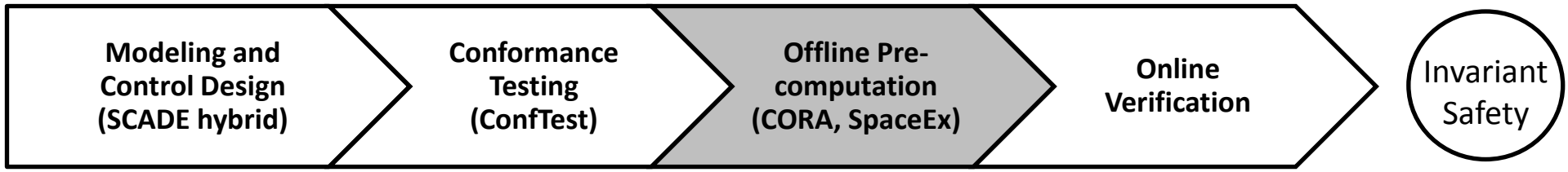
- Model $\dot{x} = f(x, u, e)$
- Low-level control $u = c\big(x, x^*(t)\big)$ → Trajectory tracking

DESIGN PROCESS



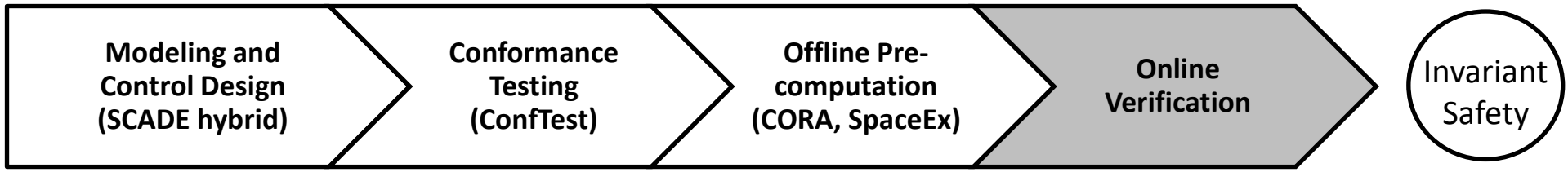
- Validate behavior of model f against physical system
- Error set E, which explains all observed physical behaviors with differential inclusion $\dot{x} \in \{f(x, u, e) \mid e \in E\}$

DESIGN PROCESS



- Deterministic control of the non-deterministic model/system
- Puzzle pieces from which to construct emergency maneuvers

DESIGN PROCESS

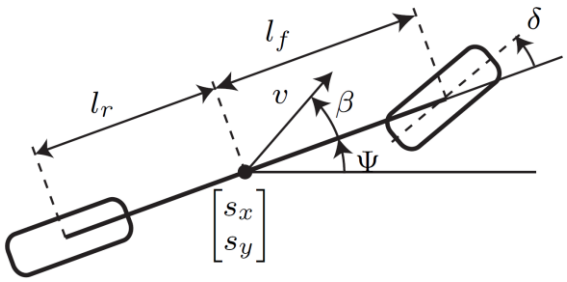


- Compute emergency maneuver from puzzle-pieces
- Validate against worst-case predictions of other traffic participants
- Validate safety-critical C2C messages
- Execute nominal maneuvers if possible
- Otherwise use „backup“ emergency maneuver

CONFORMANCE

Conformance is all about the **relationship** between the **behaviors** of an ***abstract model*** of a system and a ***reference system***.

- Refined Model
- Physical System



Abstract bicycle model

Transference of results

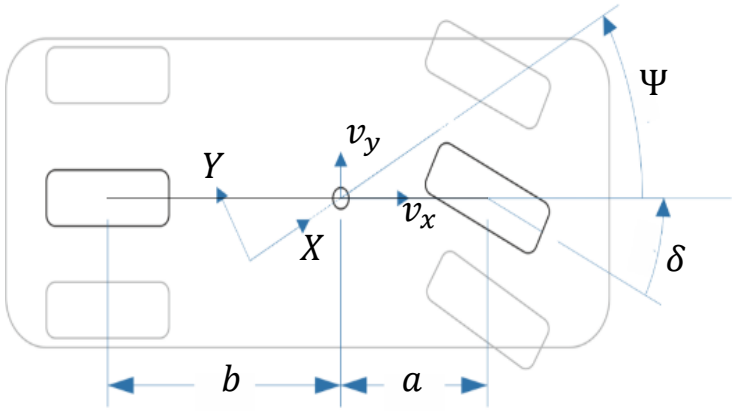
Behavior Inclusion

- Verification relies on an abstract model of an actual real system
- UnCoVerCPS MDB for AD: online decision making with safety guarantees

MDB: maneuver data base

CLOSED-LOOP ABSTRACT VEHICLE MODEL

- Abstract vehicle with states $[X, Y, \Psi, v_x, v_y, \omega]$
  - Position X,Y
  - Velocities v_x, v_y
  - Orientation Ψ and yaw-rate ω



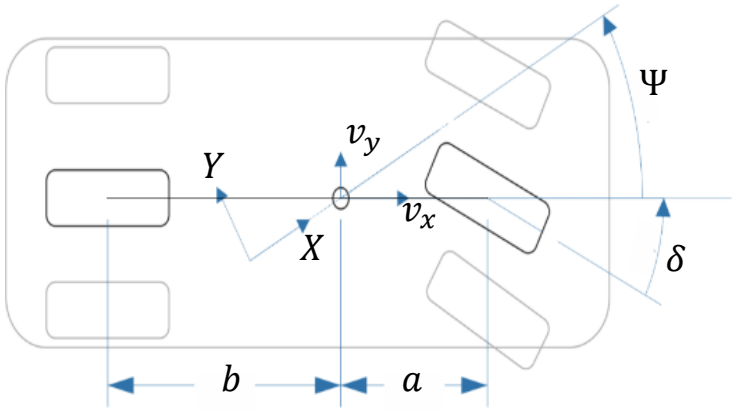
$$\dot{X} = v_x \cos(\Psi) - v_y \sin(\Psi) \oplus [-\boldsymbol{U_x}, \boldsymbol{U_x}]$$
$$\dot{Y} = v_x \sin(\Psi) + v_y \cos(\Psi) \oplus [-\boldsymbol{U_y}, \boldsymbol{U_y}]$$
$$\dot{v}_x = u_1 + v_y \omega$$
$$\dot{v}_y = f_{y,f}(x,u) + f_{y,r}(x) - v_x \omega - b\dot{\omega}$$
$$\dot{\Psi} = \omega \oplus [-\boldsymbol{U_\Psi}, \boldsymbol{U_\Psi}]$$
$$\dot{\omega} = a\frac{m}{J} f_{y,f}(x,u) - b\frac{m}{J} f_{y,r}(x)$$

\+ non-linear controller

$\oplus$: Minkowski sum of two set: $A \oplus B = \{ a + b \mid a \in A, b \in B \}$

CLOSED-LOOP ABSTRACT VEHICLE MODEL

- Abstract vehicle with states $[X, Y, \Psi, v_x, v_y, \omega]$
  - Position X,Y
  - Velocities v_x, v_y
  - Orientation Ψ and yaw-rate ω



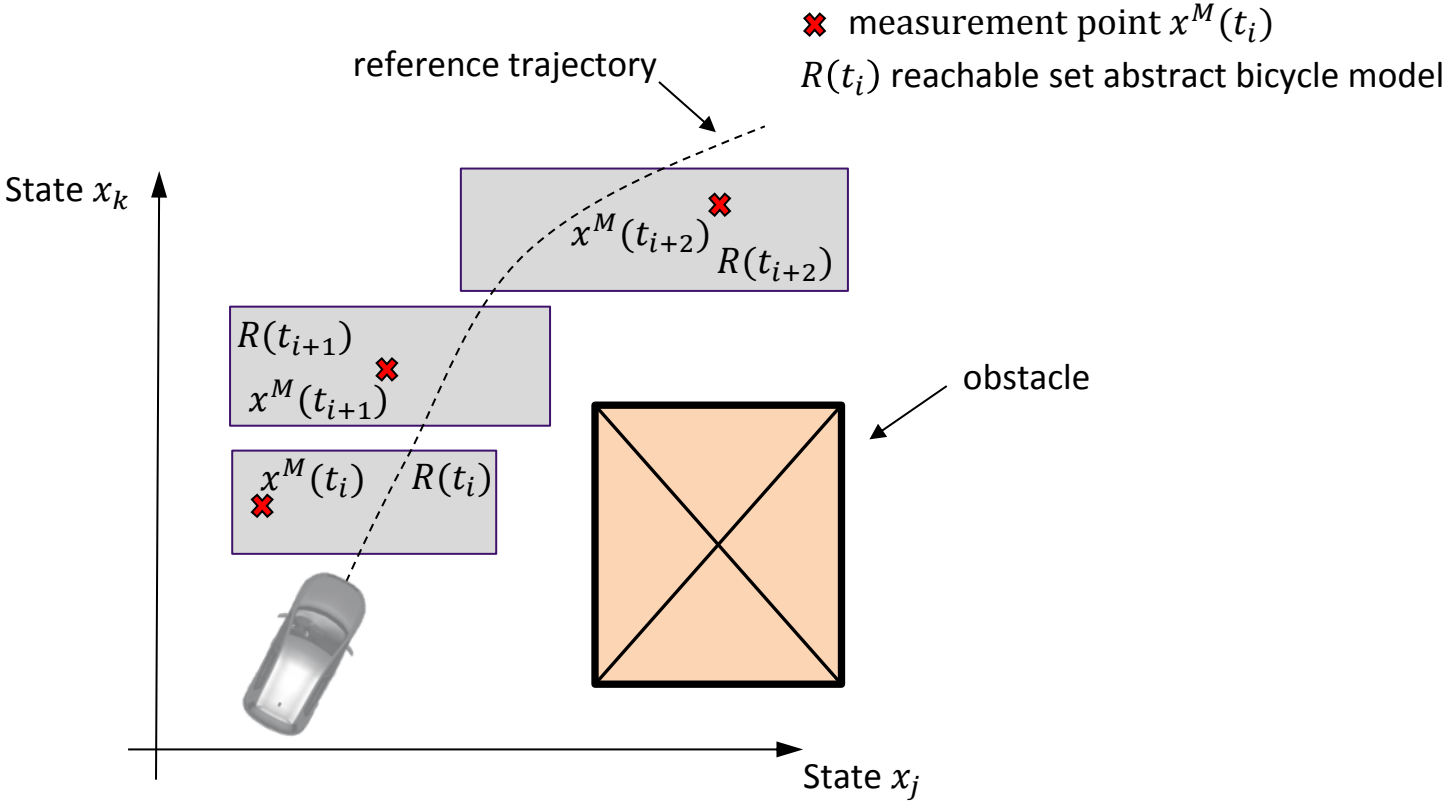
$$\dot{X} = v_x \cos(\Psi) - v_y \sin(\Psi) \oplus [-\boldsymbol{U_x}, \boldsymbol{U_x}]$$
$$\dot{Y} = v_x \sin(\Psi) + v_y \cos(\Psi) \oplus [-\boldsymbol{U_y}, \boldsymbol{U_y}]$$
$$\dot{v}_x = u_1 + v_y \omega$$
$$\dot{v}_y = f_{y,f}(x,u) + f_{y,r}(x) - v_x \omega - b\dot{\omega}$$
$$\dot{\Psi} = \omega \oplus [-\boldsymbol{U_\Psi}, \boldsymbol{U_\Psi}]$$
$$\dot{\omega} = a\frac{m}{J} f_{y,f}(x,u) - b\frac{m}{J} f_{y,r}(x)$$

\+ non-linear controller

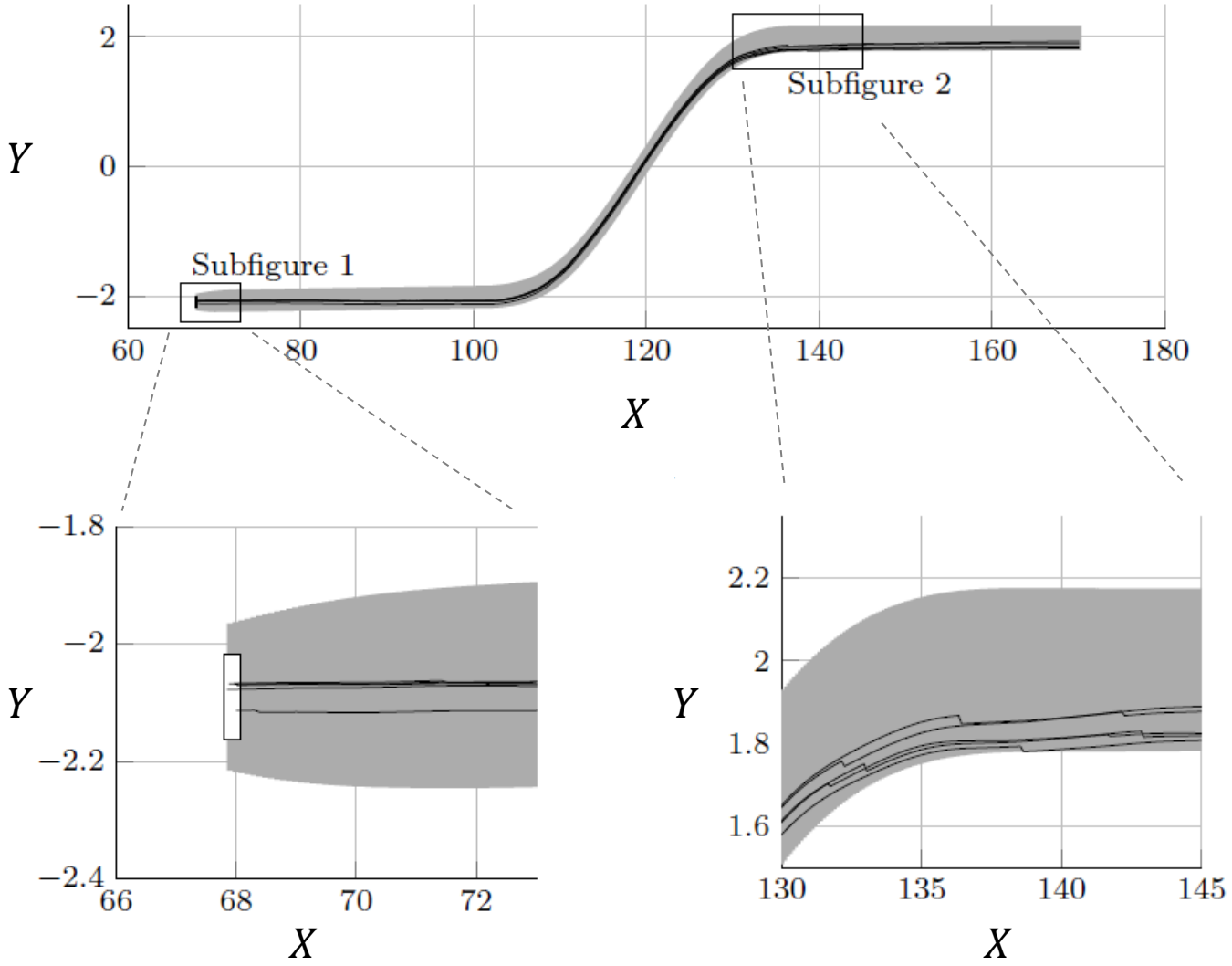
$\oplus$: Minkowski sum of two set: $A \oplus B = \{ a + b \mid a \in A, b \in B \}$

CHECKING REACHSET CONFORMANCE

- Recorded measurement data for several maneuvers with DLR vehicle
- Reachset conformance for safety properties, e.g., reach-avoid

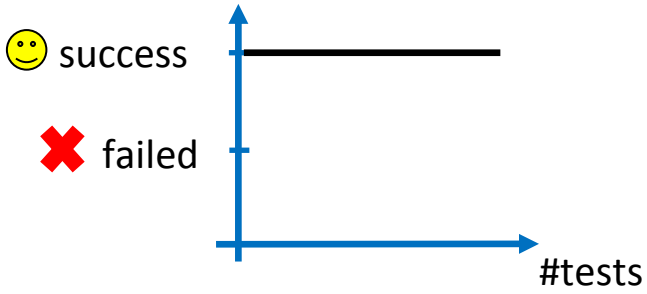


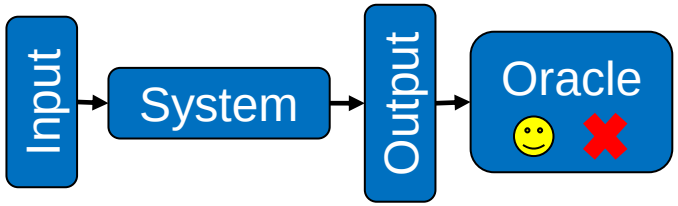
CONFORMANCE CHECKING RESULTS



CONFORMANCE TESTING

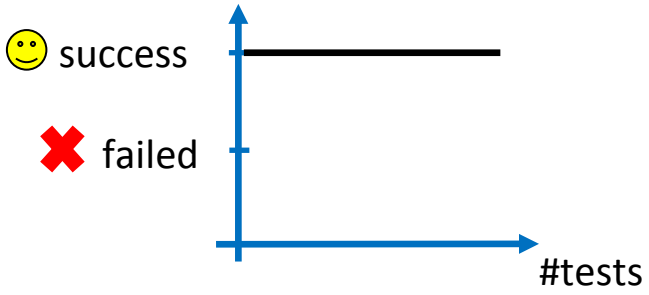
- Black box testing á la *S-Taliro* (Fainekos et al.) and *Breach* (Donzé et al.)
- #failed/successful test vs. quantitative metric of requirement violation

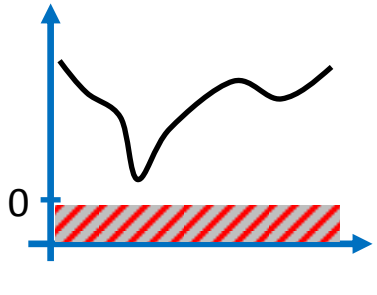




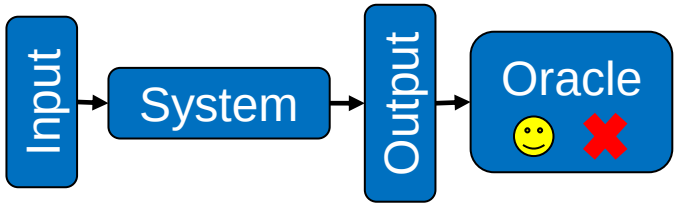
CONFORMANCE TESTING

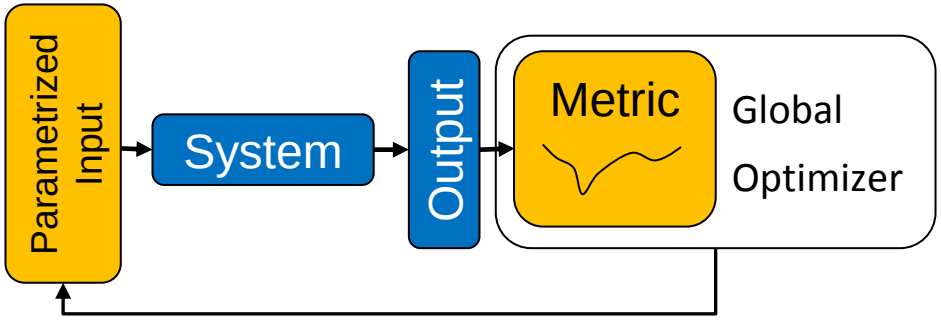
- Black box testing á la *S-Taliro* (Fainekos et al.) and *Breach* (Donzé et al.)
- #failed/successful test vs. quantitative metric of requirement violation



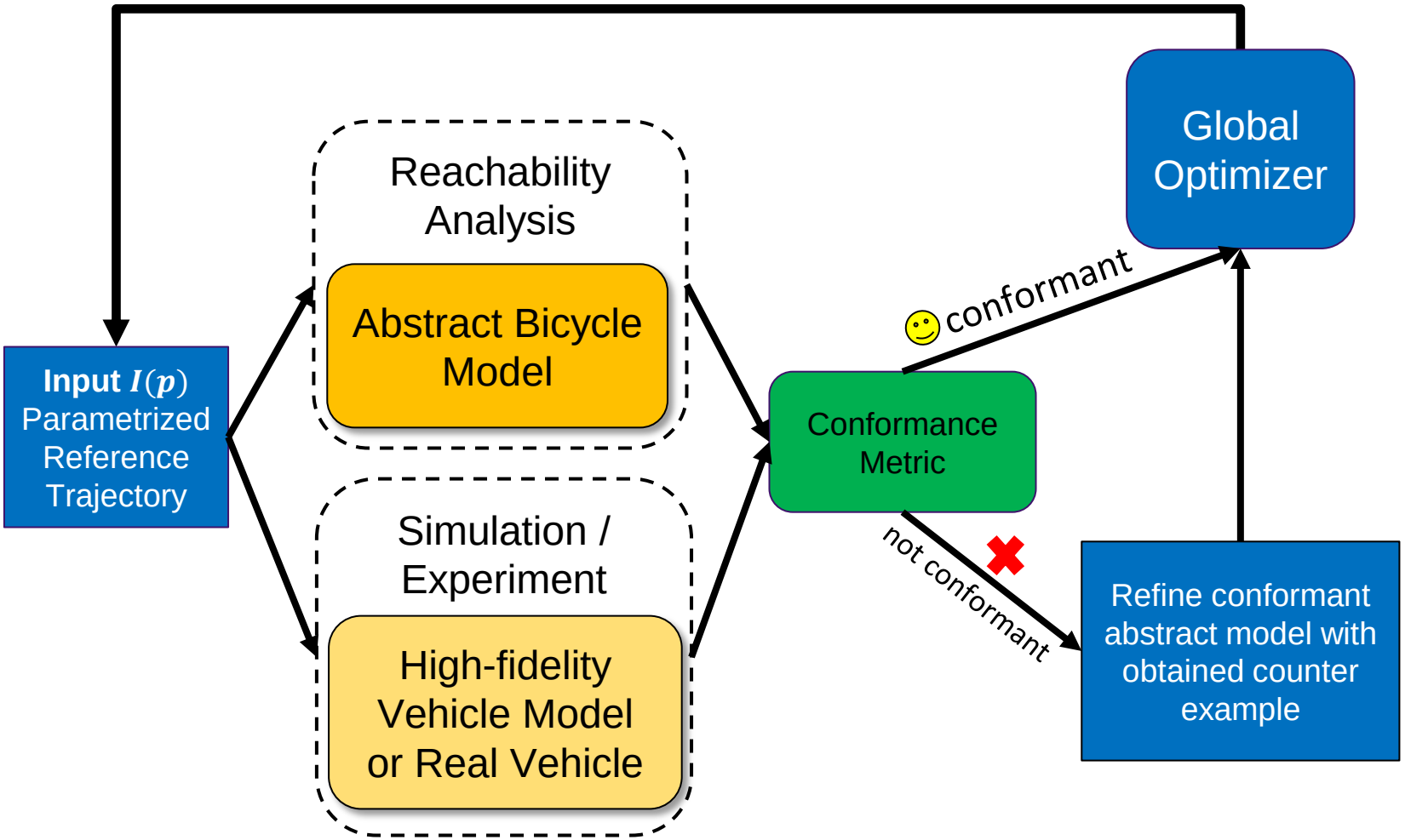


+ global optimization
  - Stochastic Optimization
  - Metaheuristics



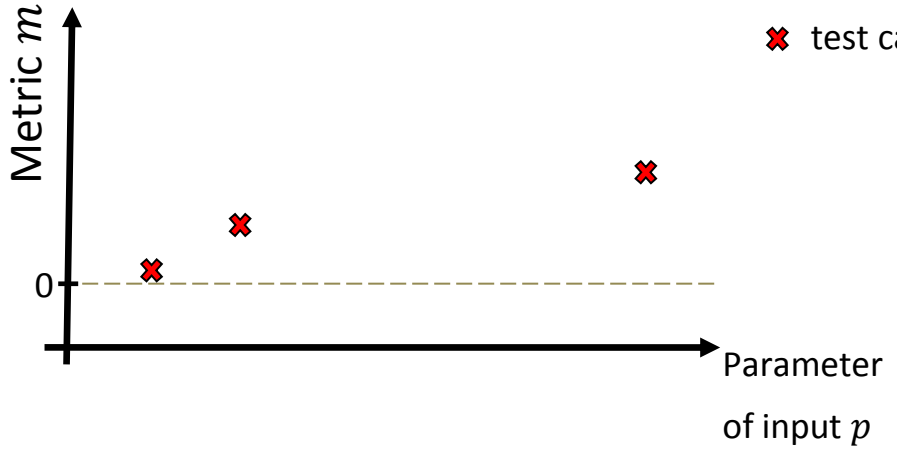


CONFORMANCE TESTING LOOP



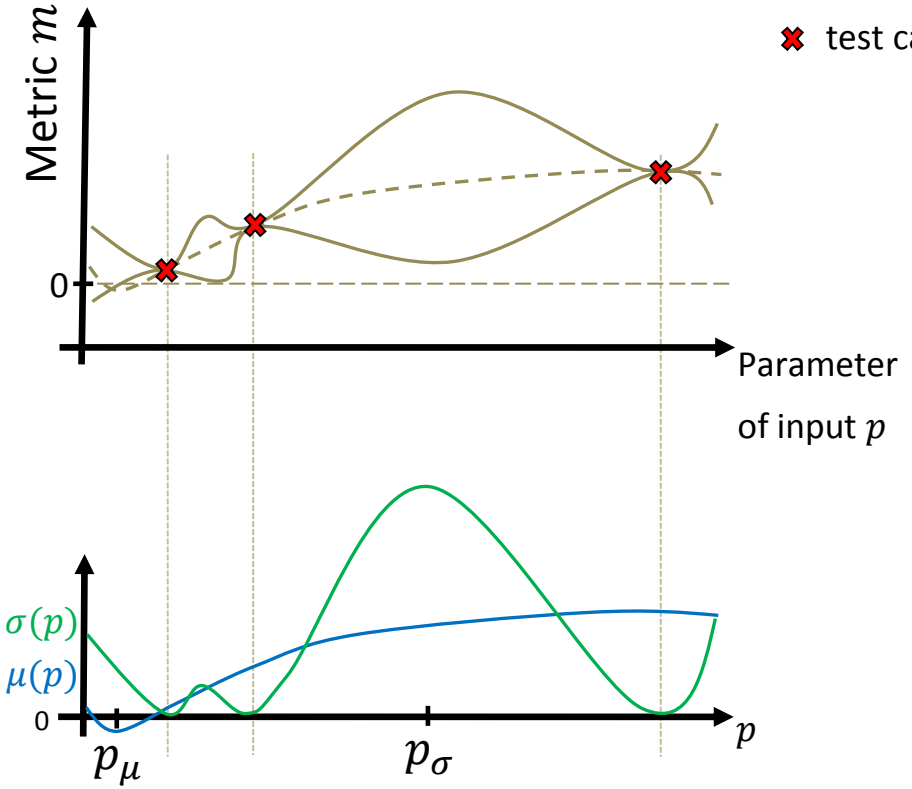
COVERAGE AS TEST END CRITERION

- Testing → non-local faults in your implementation (no singularities)
- Cannot test your whole parameter space → generalize available test database
- Related work on CPS testing via Bayesian optimization by Deshmukh et al.



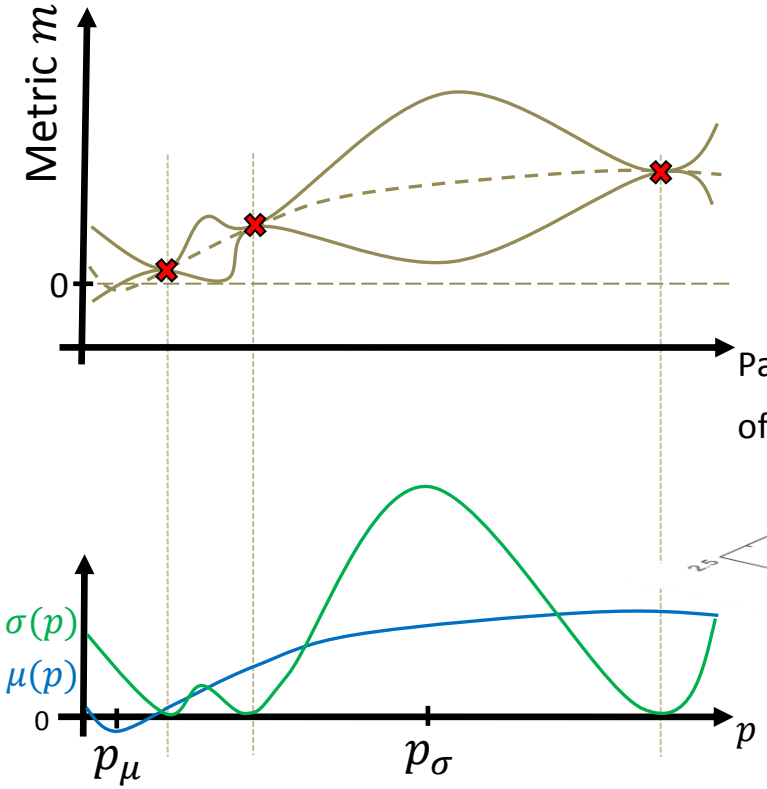
COVERAGE AS TEST END CRITERION

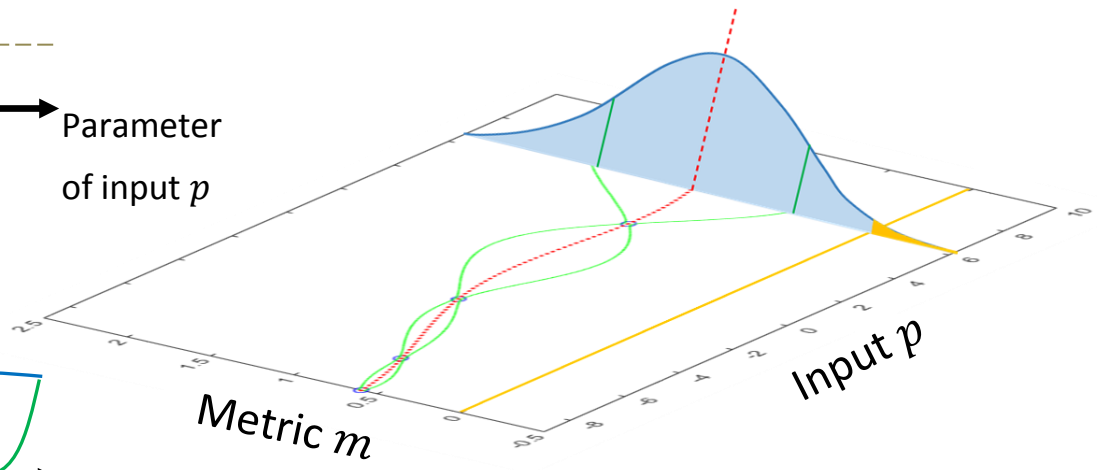
- Testing → non-local faults in your implementation (no singularities)
- Cannot test your whole parameter space → generalize available test database
- Related work on CPS testing via Bayesian optimization by Deshmukh et al.



COVERAGE AS TEST END CRITERION

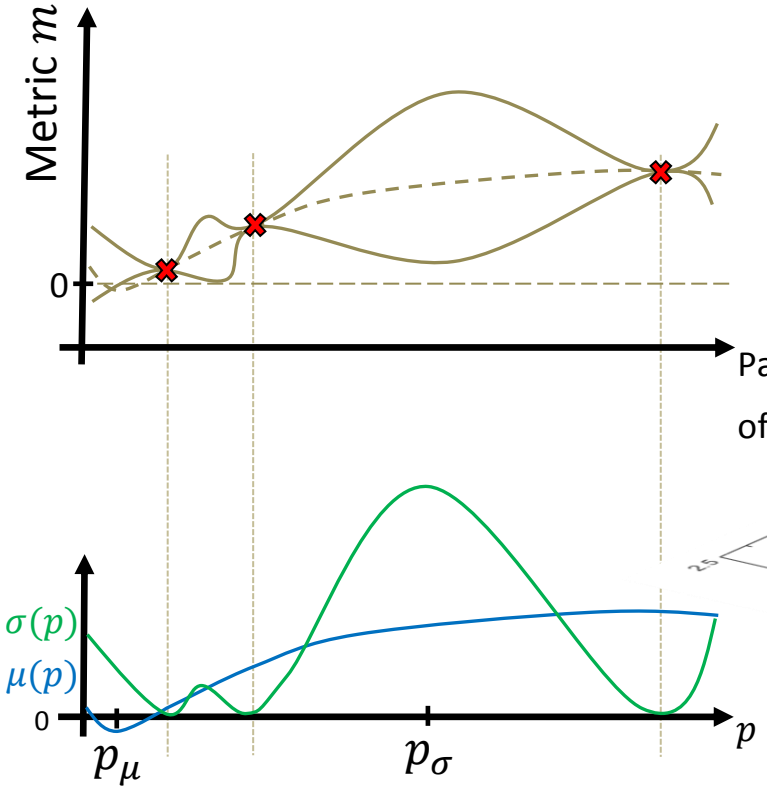
- Testing → non-local faults in your implementation (no singularities)
- Cannot test your whole parameter space → generalize available test database
- Related work on CPS testing via Bayesian optimization by Deshmukh et al.



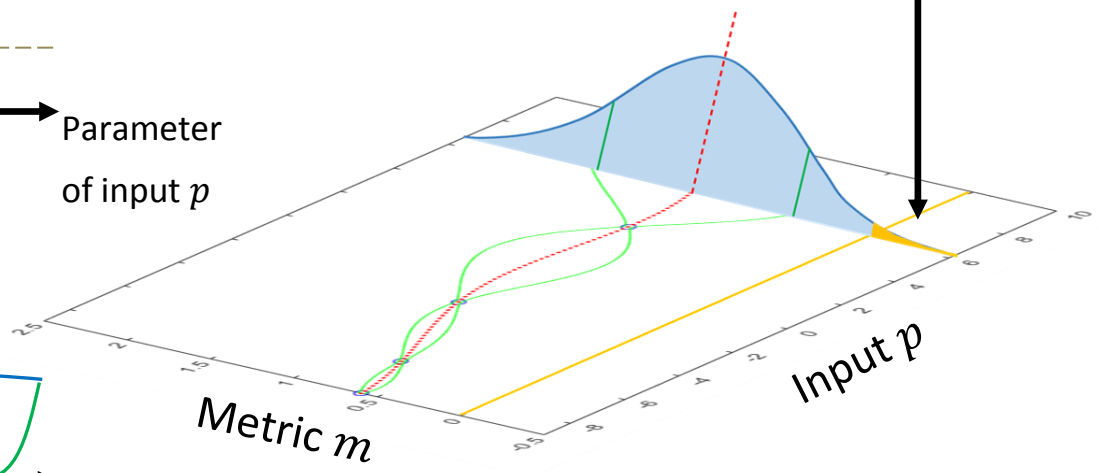


COVERAGE AS TEST END CRITERION

- Testing → non-local faults in your implementation (no singularities)
- Cannot test your whole parameter space → generalize available test database
- Related work on CPS testing via Bayesian optimization by Deshmukh et al.



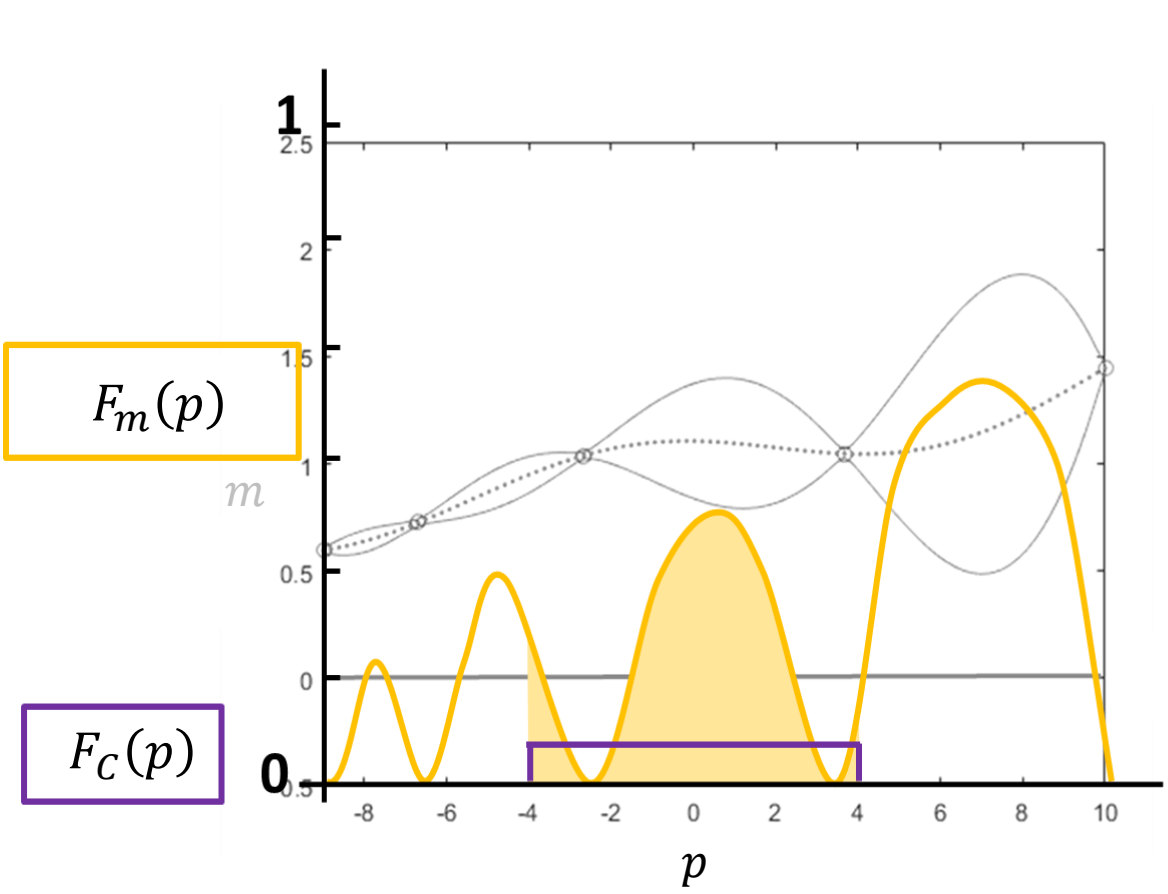
$$F_m(p) = \int_{-\infty}^{0} P(m(p))\, dm$$

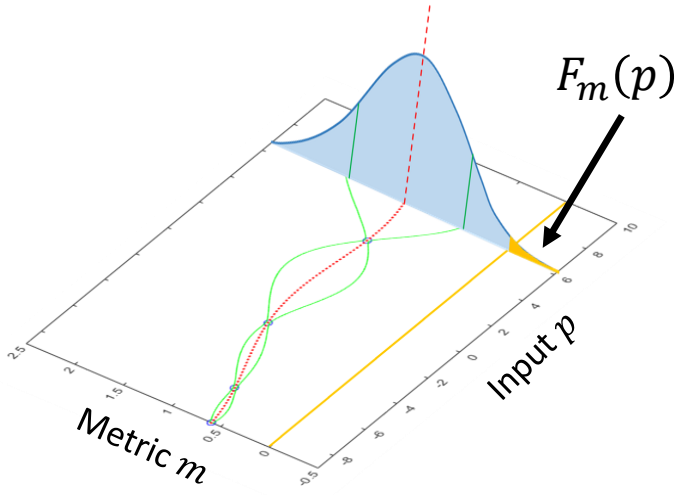


$$p_{bayes} = \underset{p}{\operatorname{argmax}}(F_m(p))$$

COVERAGE AS TEST END CRITERION

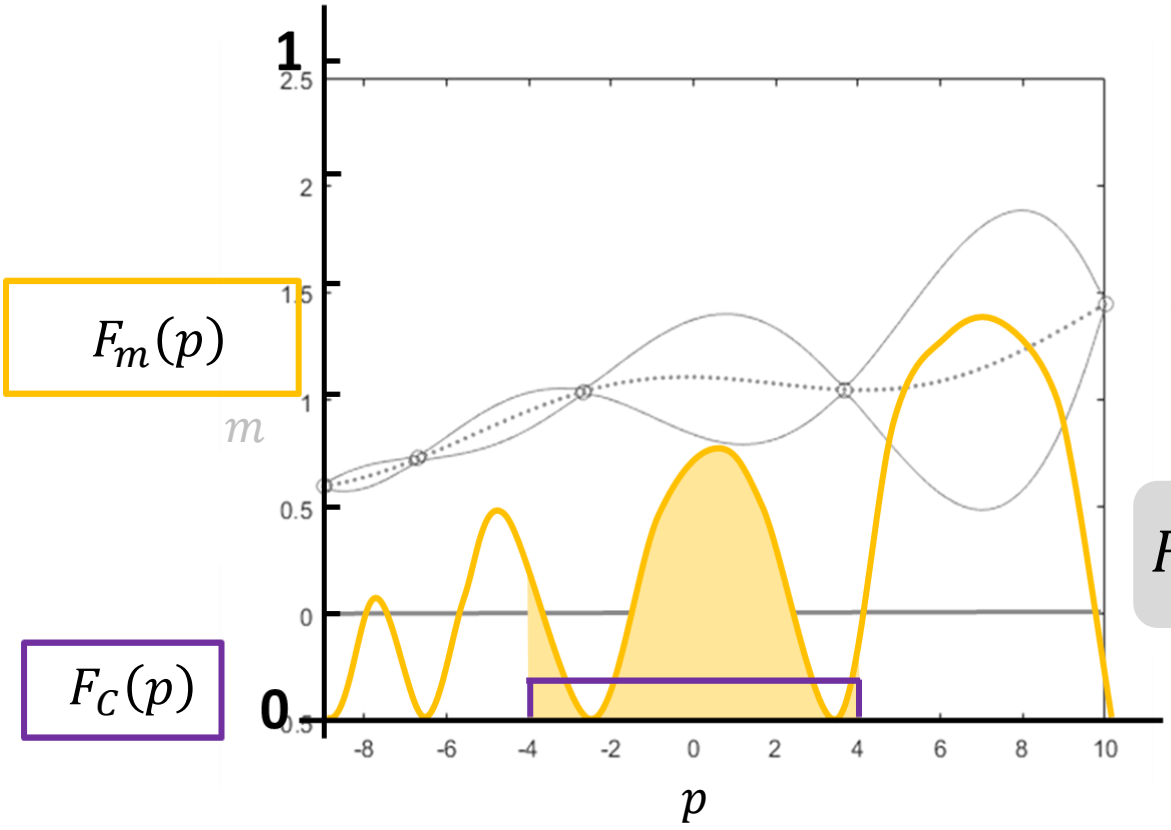
- Leverage Gaussian process as a **surrogate model for coverage**

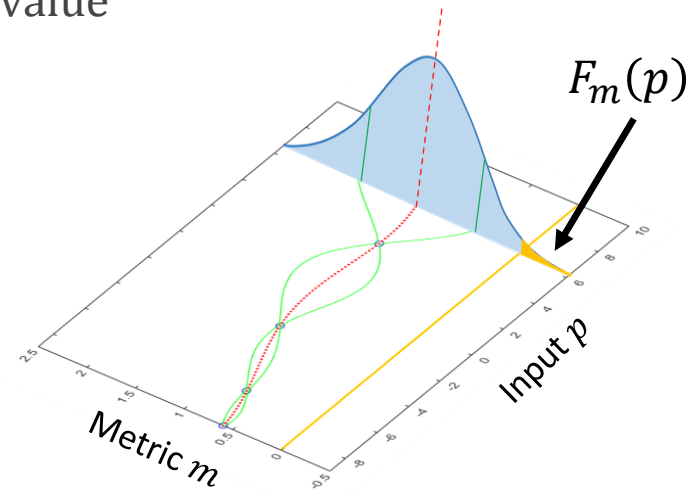




COVERAGE AS TEST END CRITERION

- Leverage Gaussian process as a **surrogate model for coverage**
- The sample space is sufficient if a randomly chosen input violates conformance with a probability less than a given value



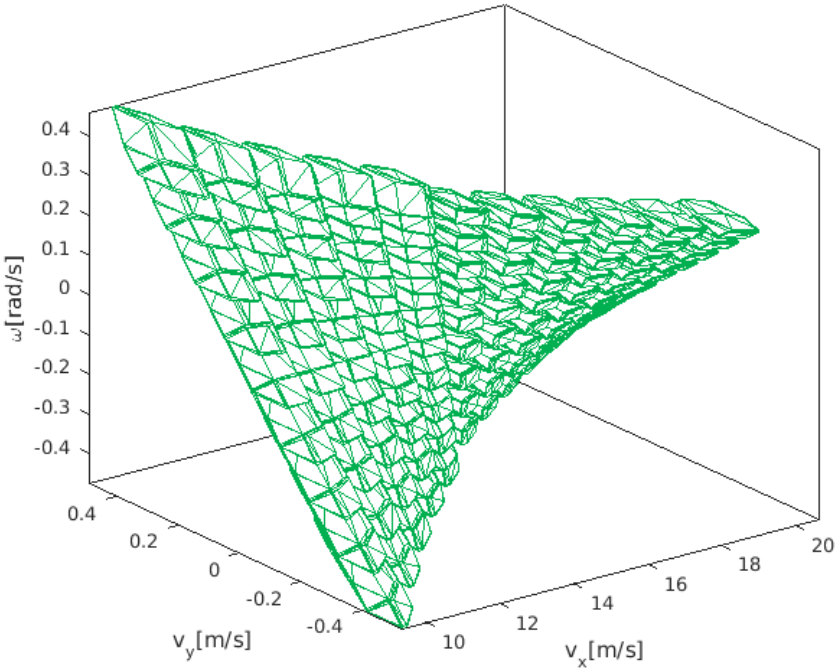


$$F_C(p) = \int_{p_{min}}^{p_{max}} F_m(p)\, dp \leq^{!} C$$

OFFLINE PRECOMPUTATION

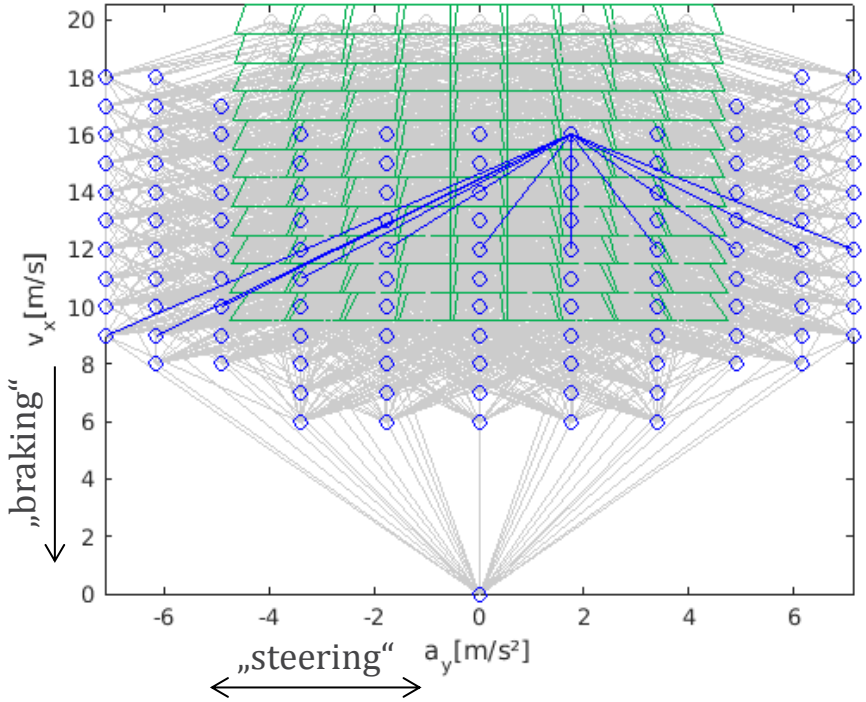
Nominal Maneuvers

Continuous coverage of reference state space near steady state driving surface



Emergency Maneuvers

Connections from nominal maneuvers to discretely sampled emergency maneuvers, as well as interconnections



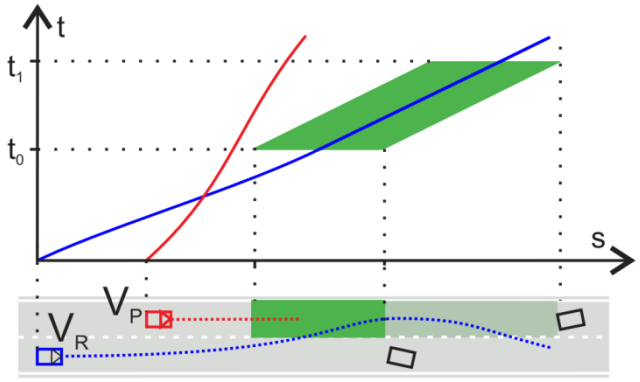
⇒ ca. 12k maneuvers total

[Heß, D.; Löper, C. and Hesse, T.: *Safe cooperation of automated vehicles*. AAET 2017]

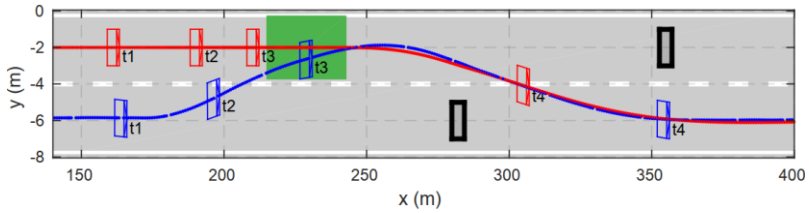
NOMINAL PLANNER

- Decoupled long. and lat. planning for 3rd order integrator chain
- Formulation of different cooperative and non-cooperative nominal maneuvers as quadratic optimization problems
- Computation of cooperative lane changing maneuvers with 10s horizon in ca. **1ms, up to 3.5ms** worst case

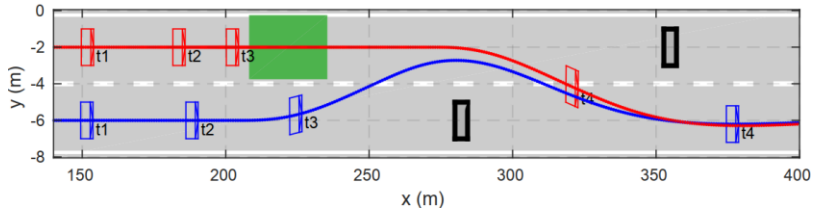
Cooperation: Space-time reservation protocol



Experiment A: Test vehicle (blue) & simulated (red)

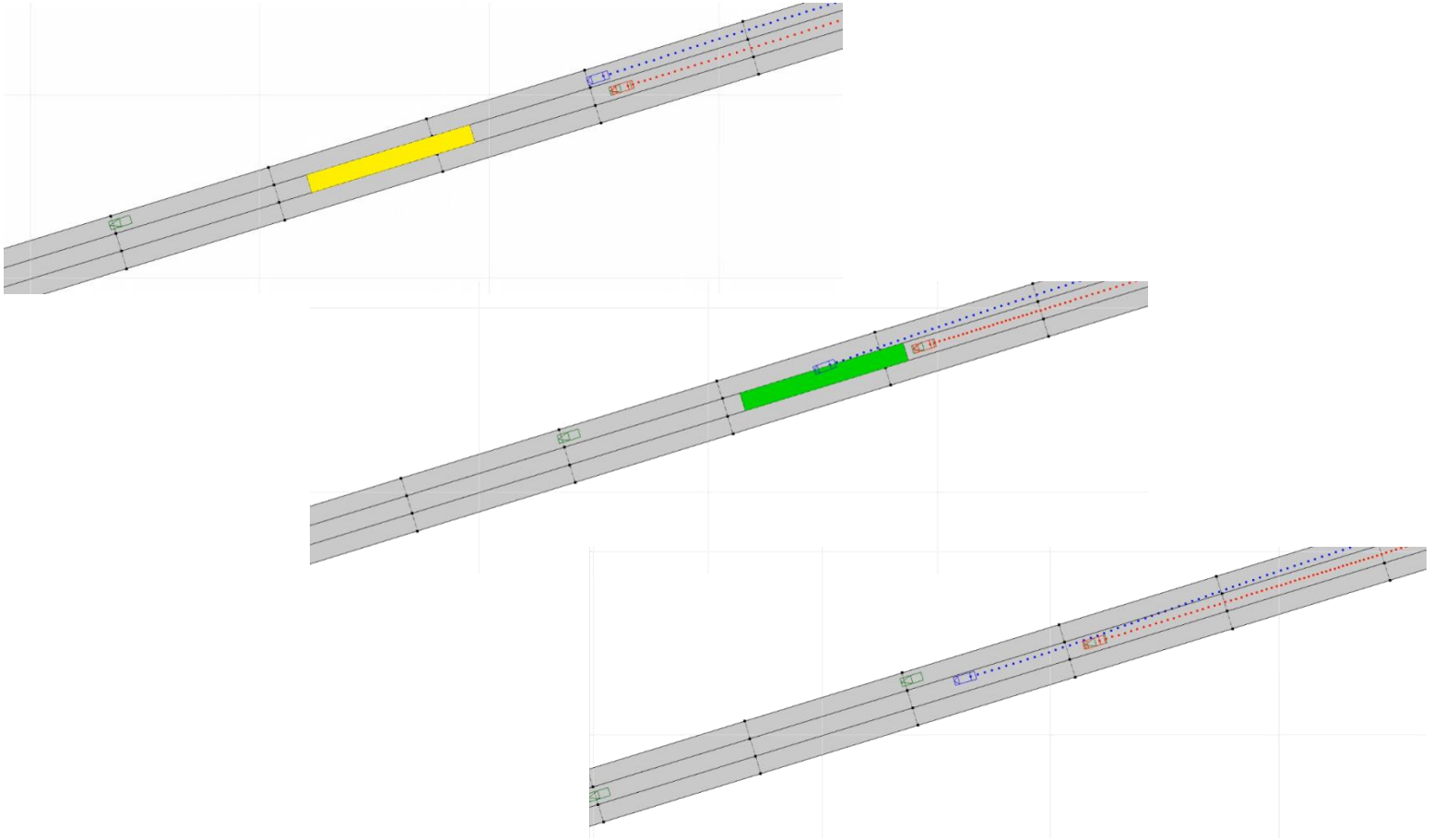


Experiment B: Both simulated



[Heß, D.; Lattarulo, et al.: *Fast maneuver planning for cooperative automated vehicles*. Submitted to ITSC'18]

RESULTS: VEHICLE COOPERATION



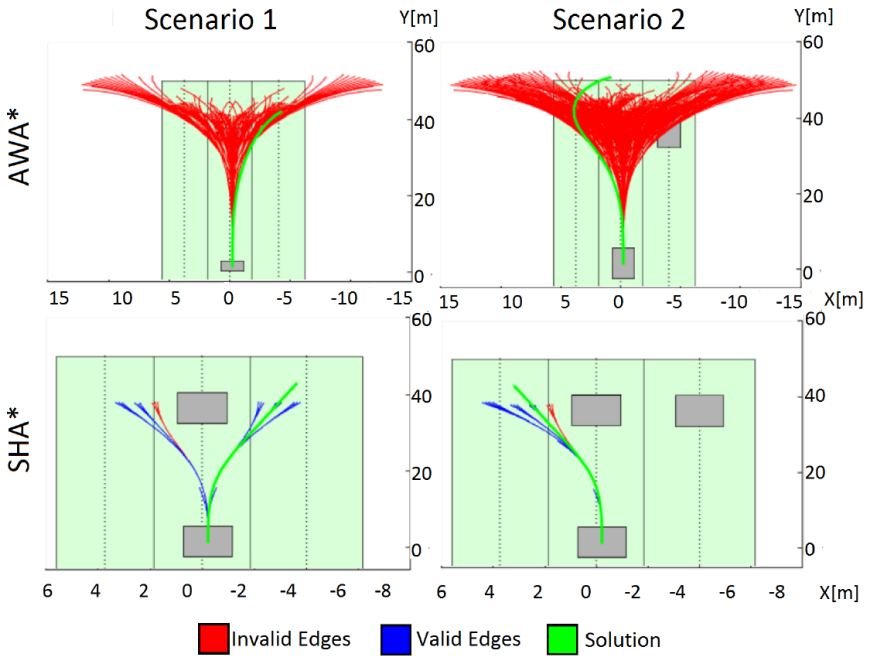
Link to the video: https://youtu.be/PuvfMMz-zM8

ONLINE VERIFICATION

- Motion planning problem: Find shortest emergency maneuver sequence leading to a safe stand-still
- Discrete sampling

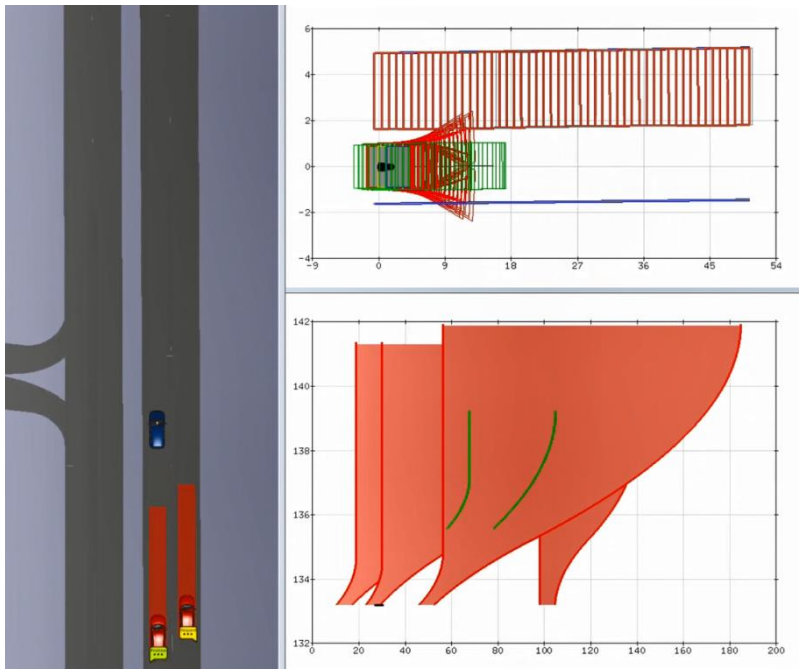
  → Graph search, A* variants
- Ca. **3 ms** computation time

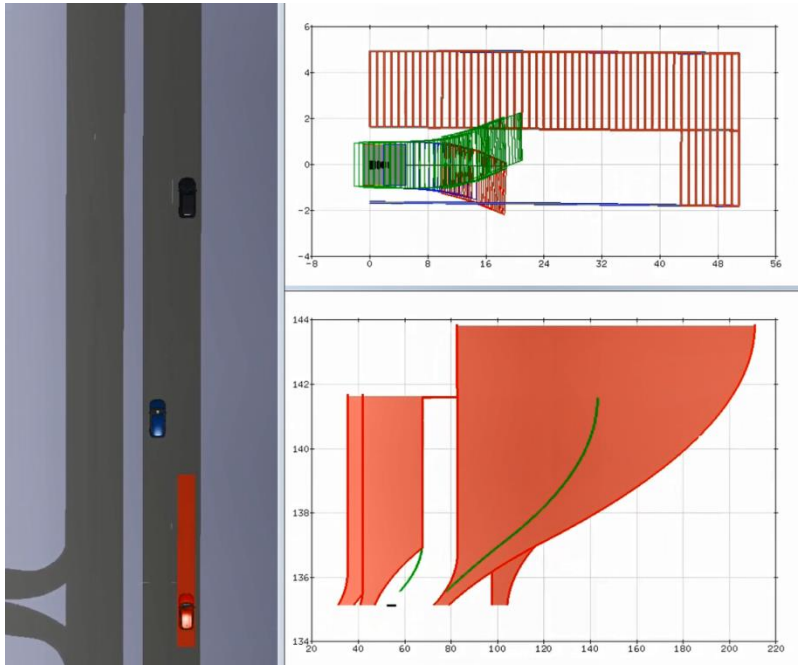
  for first “sub-optimal” solution



[Salvado,J; Custódio, L.; and Heß, D. "Contingency planning for automated vehicles IROS 2016.]

RESULTS: ONLINE VERIFICATION





Link to the video: https://youtu.be/aaHUvt_OCWU

CONCLUSION

- Invariant Safety for automated vehicles: Guarantee existence of a safe emergency maneuver → "Online Verification"
- Guaranteed over-approximation of system's behavior during nominal and emergency maneuvers with CORA

  → Guaranteed collision avoidance
- Validation of non-deterministic model: Conformance testing
  - Challenge for coverage: hyper-parameters for generalizing of test cases and test end threshold
- Pre-computation of maneuver automata for real-time performance
  - Nominal and emergency maneuver planning: Ca. 3ms each
  - Update cycle of 100ms
- General applicability as "safety layer" due to black-box assumption

PREVIEW OF FINAL DEMONSTRATION

Link to the video: https://www.youtube.com/watch?v=6JDpNR7Dpjo

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643921.

THANK YOU