Continental

The Future in Motion

# SystemC-AMS and AMESIM Cosimulation

## Virtual prototyping of a solenoid injector control of a power-train automotive system

# Motivation

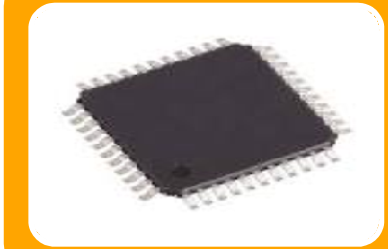
- Within automotive domain, activities are historically organized in silos
  - An entity deals with injector development
  - Another takes care of electronic control design
  - And another one is in charge of function and SW development
- Co-design is often very difficult
  - Different tooling, background, etc…
- Late detection of bugs (during integration of all parts on first real system prototype)

- => increasing need of bridging different domains around a common representative virtual prototype…
- …to achieve an affordable cost and time to market

# Injection Function Components and Simulation tool

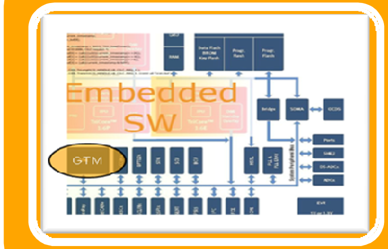
## Injector

- AMESim, …



## Asic

- SystemC AMS, …



## Micro-controller

- SystemC

# Application Description: Injector as a sensor

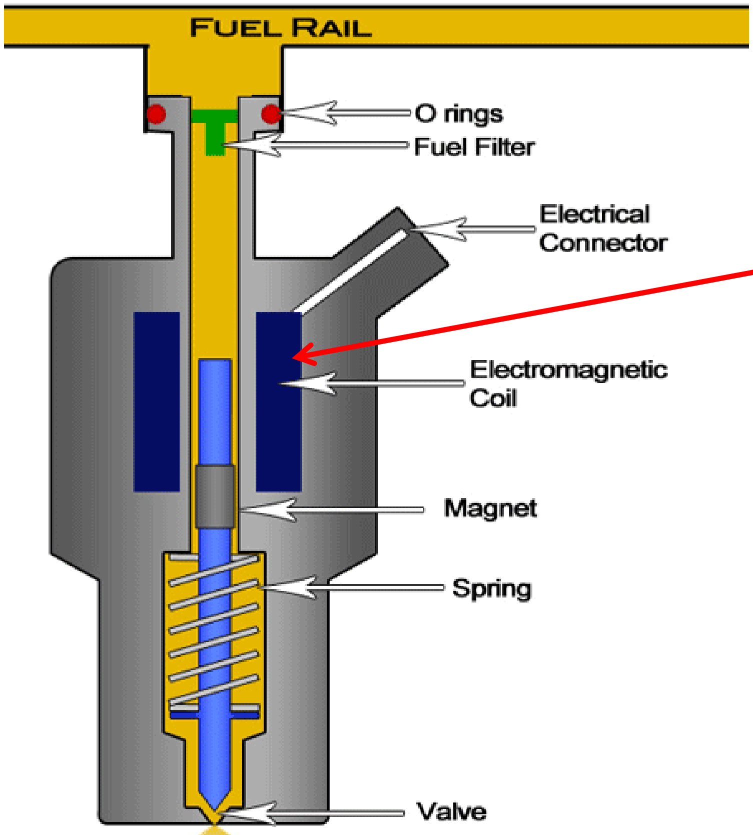
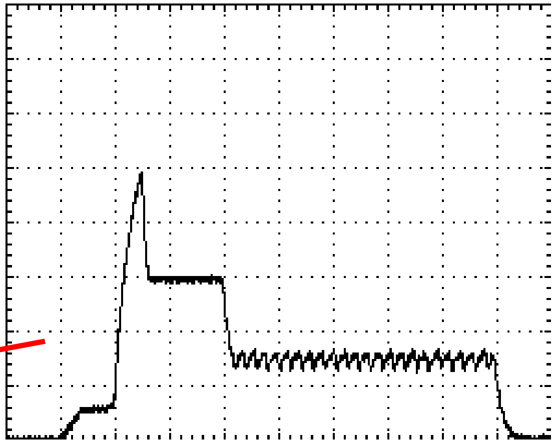


Image by William Lucas
http://www.cvel.clemson.edu/auto/AuE835_Projects_2011/Lucas_project.html



- Current monitoring is used to control injector opening
- Precise current sensing, i.e. monitoring needle movement effects on the current could allow compensation of injector ageing or injector discrepancy from one to the other

# Designing a close interacting « cross-domain » feature

- Very painful today
  - Test results are late (integration tests)
  - Many re-design loops
- Ex1: Microcontroller resources shortage => ASIC should be extended
- Ex2: Injector feedback signal too small => coil or needle update...

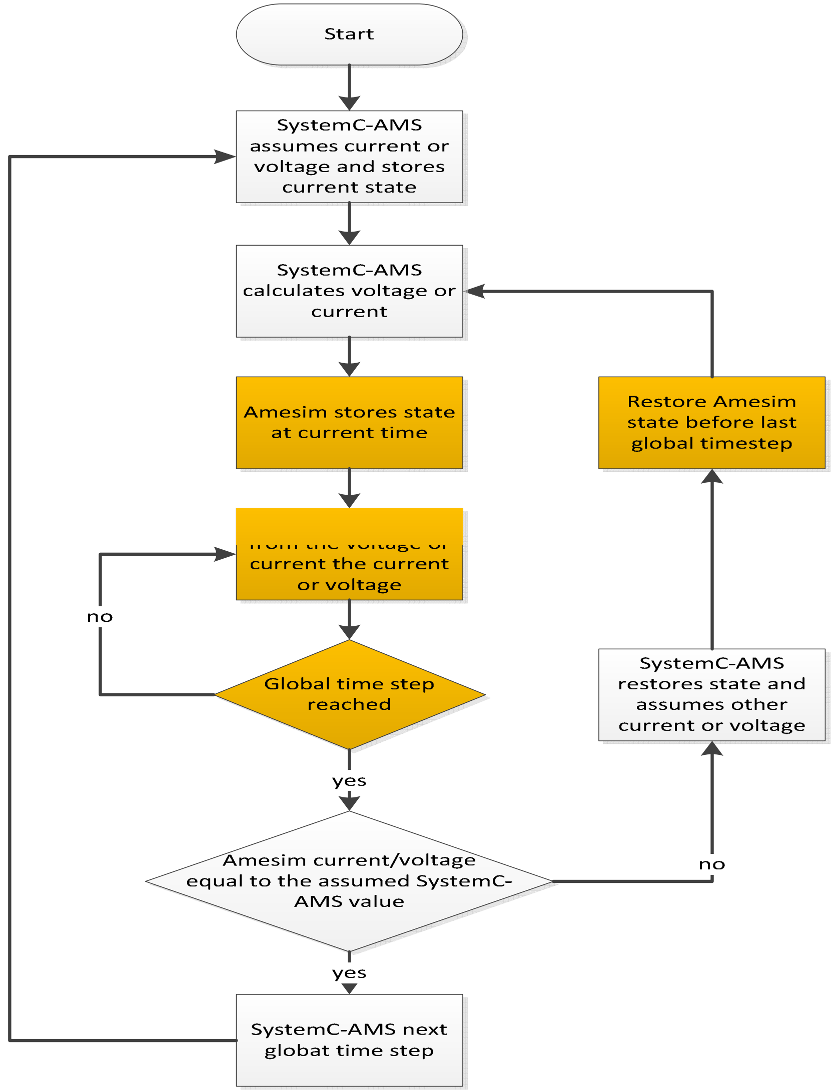
- => A virtual prototype would allow
  - Close cross-domain cooperation
  - Early integration tests
- => best design of mechanic, control electronics and SW
- Co-simulation is state-of-the-art

# Expectations

| | Today | Expected gain |
|---|---|---|
| Dev Cost | SW, ASIC or injector are re-designed for each major correction | Right the first time<br>› No added ASIC Sample (metal mask)<br>› Less SW/Calibration releases<br>› Less Injector prototypes |
| Product Development Time | Each new component release takes time | Right the first time<br>› No delay |
| Performance (“BOM Cost”) | Average | Optimum |

“performance” means achieving best Build of Material cost to fulfill emissions requirements and other regulations.

# AMESim and SystemC AMS coupling



Start

SystemC-AMS assumes current or voltage and stores current state

SystemC-AMS calculates voltage or current

Amesim stores state at current time

from the voltage or current the current or voltage

Global time step reached

no

yes

Amesim current/voltage equal to the assumed SystemC-AMS value

yes

SystemC-AMS next globat time step

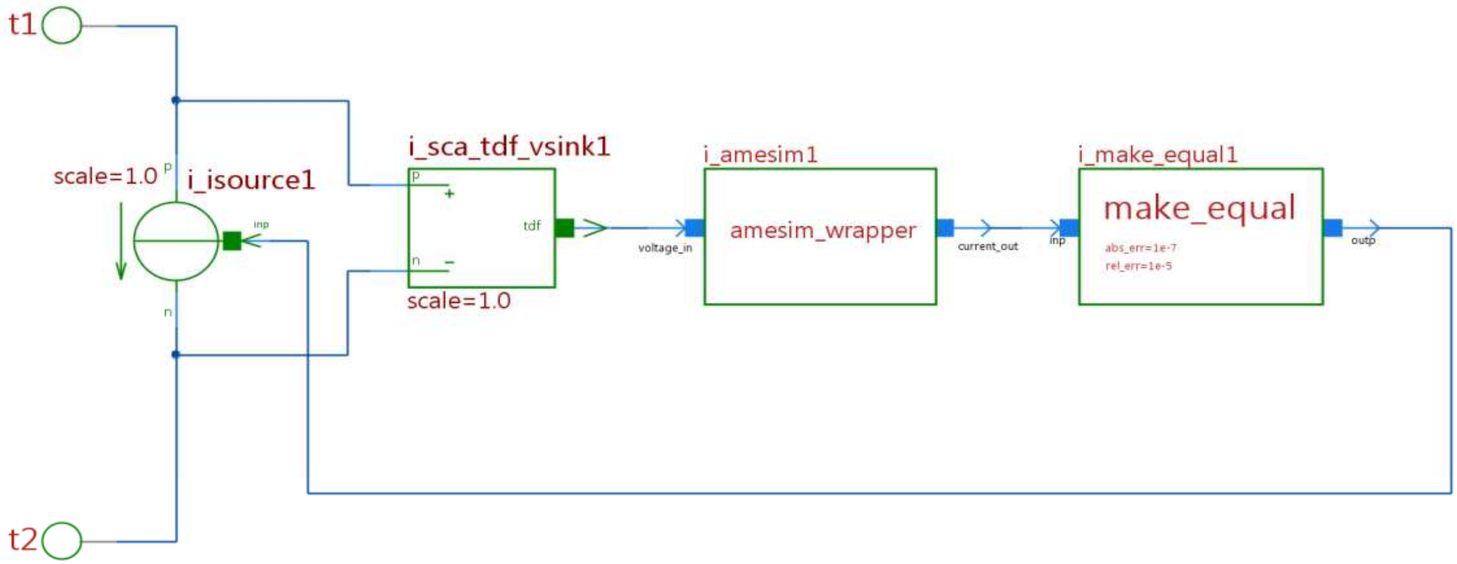
no

SystemC-AMS restores state and assumes other current or voltage

Restore Amesim state before last global timestep

- AMESim iterates with assumed current value in SystemC AMS
- SystemC AMS checks assumed AMESim value matches expected value
- If error is too large, process is re-started with a new value

# AMESim and SystemC AMS coupling



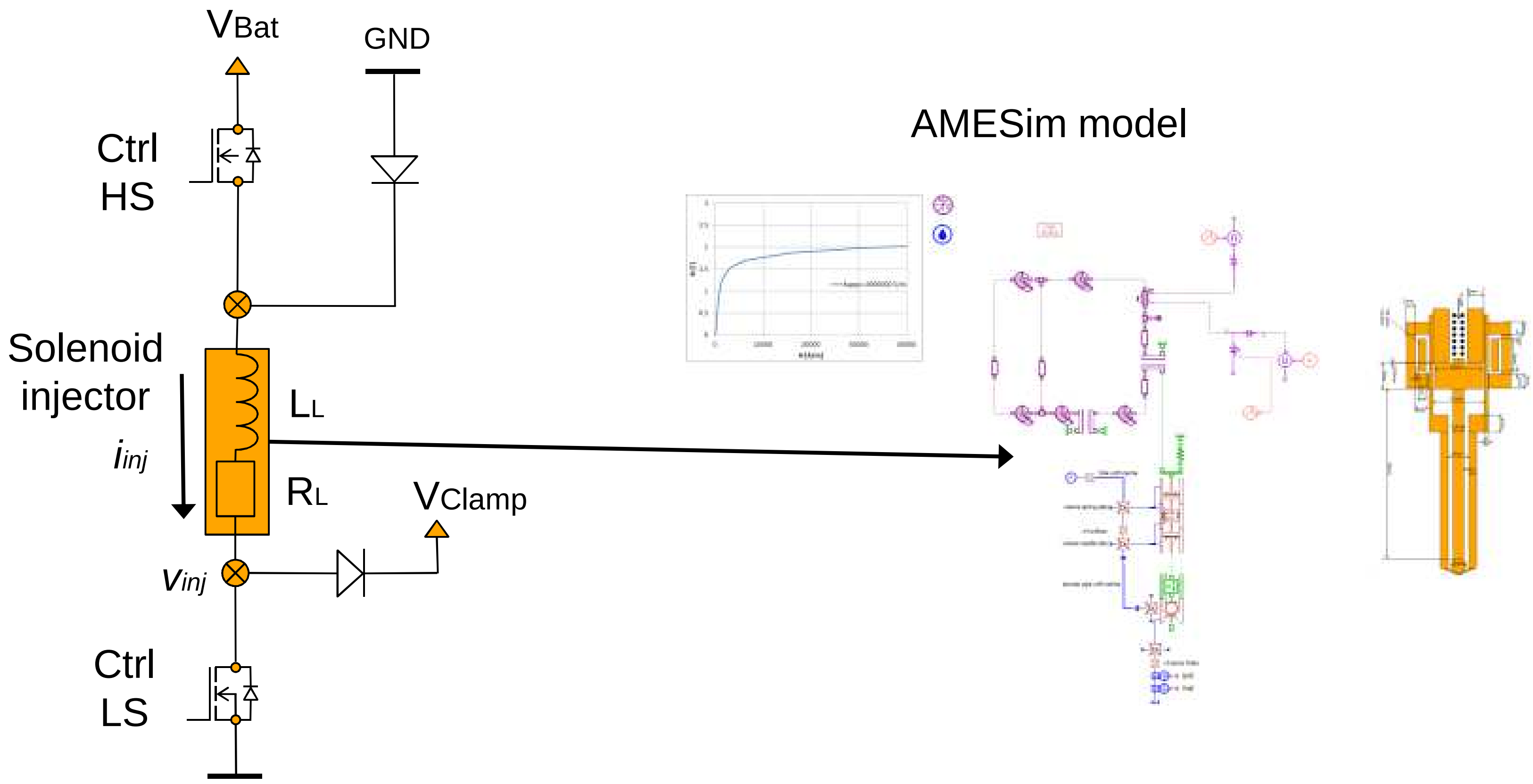
- *make_equal* assumes a current
- SystemC AMS computes a voltage, provided to AMESim
- AMESim calculates a current with this voltage
- *make_equal* compares current with assumed current and maybe repeat timestep

# Principle timestep repeat with SystemC-AMS 2.0

```
void make_equal::processing()
{
  if((fabs(current_inp.read()-current_out_last)>max_error)
  {
    result_wrong=true;
    current_out_last = …; //new assumption for the current current
  }
  else
  {
    result_wrong=false;
    current_out_last = …; //assumption for next current value
  }
  current_outp.write(current_out_last);
}

void make_equal::change_attributes()
{
  if(result_wrong)
  {
    //invalid / repeat last timestep
    request_next_activation(sc_core::SC_ZERO_TIME);
  }
}
```

# Injector Control electrical Schematic

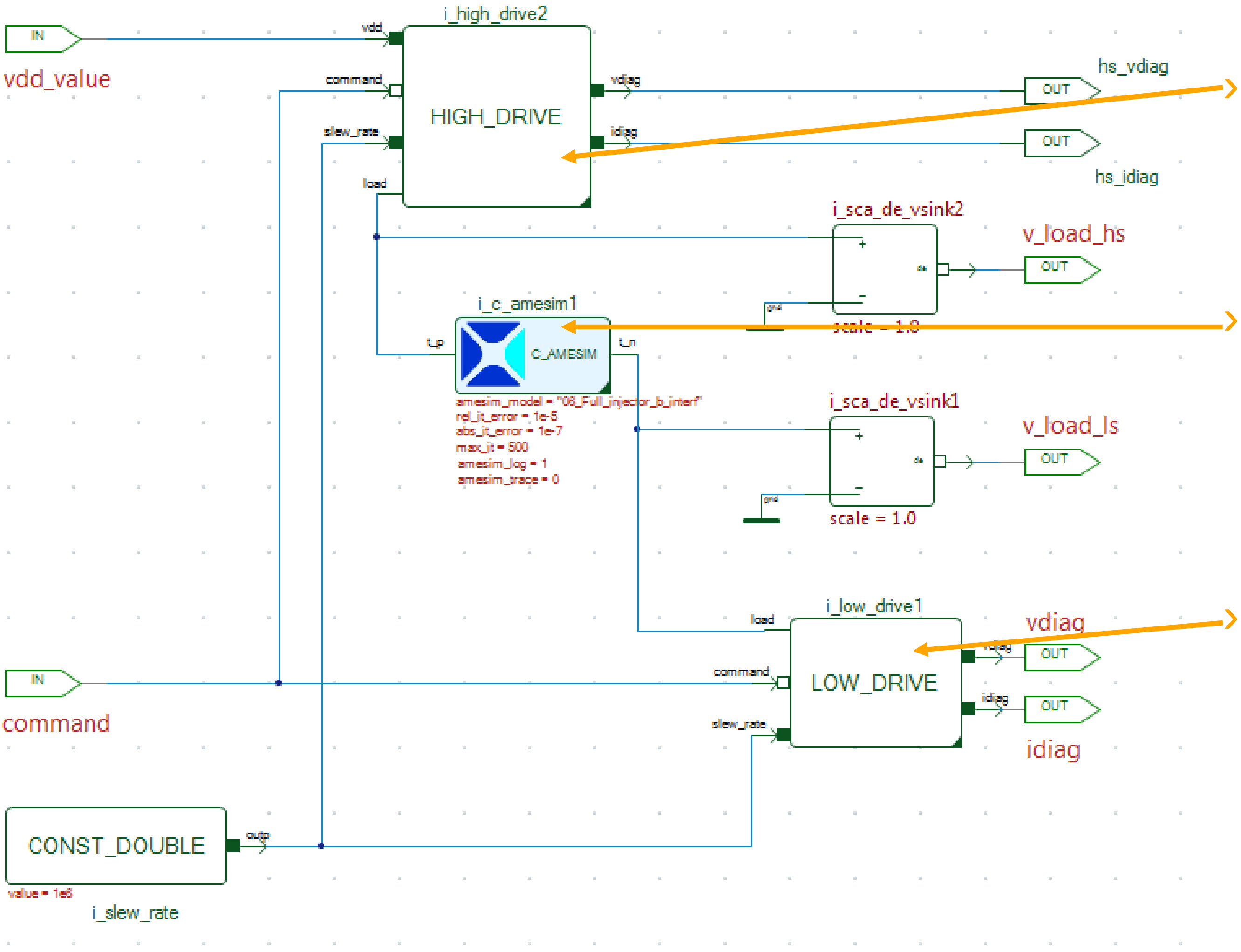


$V_{Clamp}$ = 65V

$V_{bat}$ = 15V

$R_{DSON}$ = 1Ω

# COSIDE® Top level view



amesim_model = "08_Full_injector_b_interf"
rel_it_error = 1e-5
abs_it_error = 1e-7
max_it = 500
amesim_log = 1
amesim_trace = 0

scale = 1.0

scale = 1.0

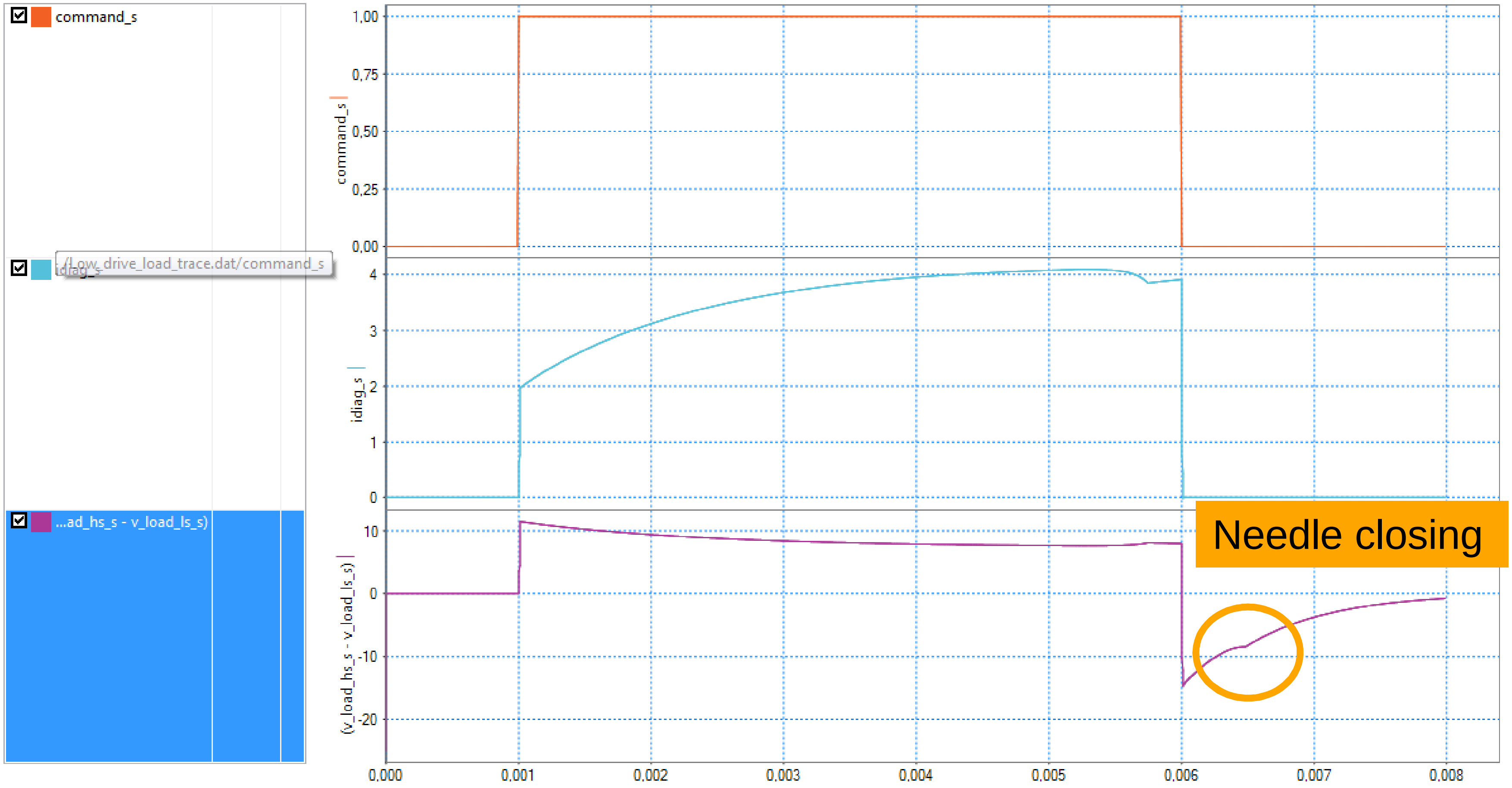
value = 1e6

High Side

TDF module encapsulating AMESim model => injector

Low Side

# Results



Needle closing

Elapsed time :133127 ms (Core i5 laptop)
Simulated time :8 ms

# Summary

- System design across domains becomes more and more essential and is enabler for a further system optimization
- AMESIM – SystemC AMS coupling enables algorithm development for thightly interacting analog digital hard-/software and mechanical components on a high accuracy level
- The simulation performance is acceptable, due the moderate number of iterations (the injector has an inductor like behaviour – so the current changes slowly and continously)
- The AMESIM SystemC AMS coupling enables a thighter cooperation of different groups
- Electronic and SW development gets closer to physics

# H-Inception

- SystemC AMS extension allowing Mutli-Domain Virtual Prototyping

- This co-simulated system reused as a reference basis within H-Inception
  - Same injector modeled within SystemC MDVP
  - Used to debug SystemC MDVP add-ons
  - And for performance comparison