Modeling an Electronic Gas/Acceleration Pedal

Anton Pirker-Frühauf
KAI-Kompetenzzentrum Automobil- und Industrieelektronik

Karsten Schönherr
Robert Bosch GmbH

Arnaud Laroche
Georg Pelz
Infineon Technologies AG

BMAS 2007-09-20

BOSCH

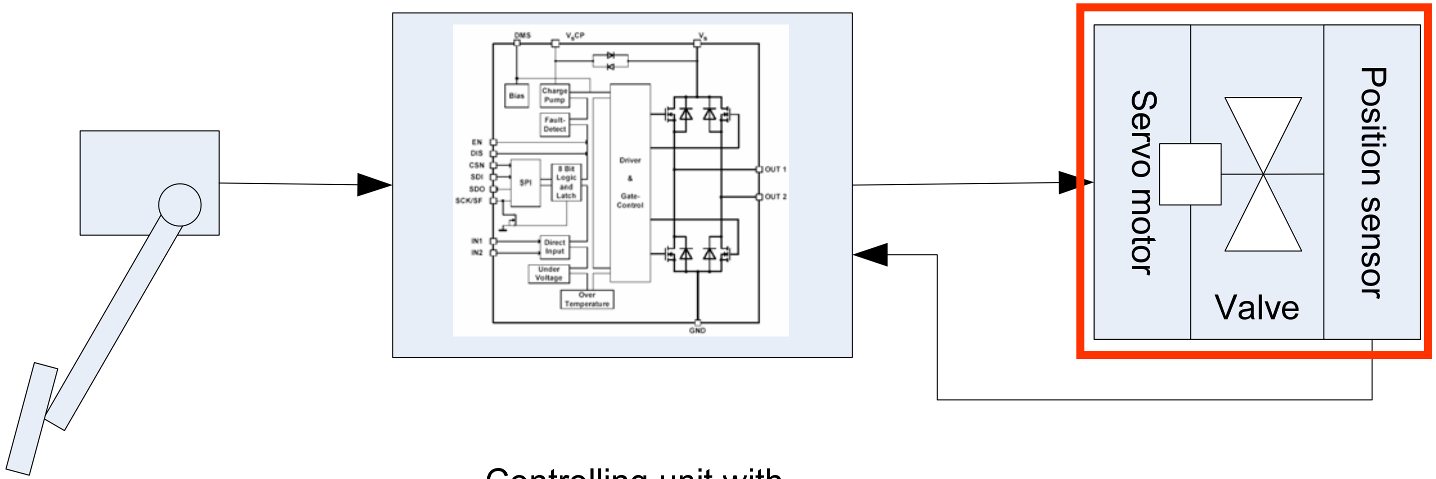
Infineon
Never stop thinking

KAI

Overview

- Introduction to electronic Gas pedal
- Throttle body components
- Compare throttle body model Matlab Simulink vs. VHDL-AMS
- VHDL-AMS code example
- Measurements
- Comparing measurement with simulation results
- Controlling unit (worst-case modeling)

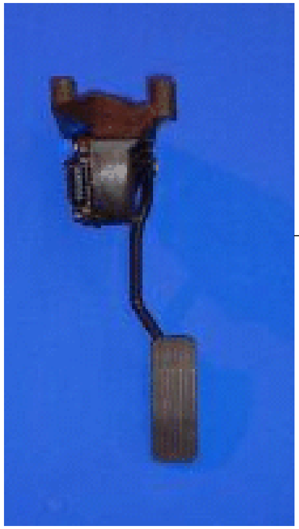
Introduction to eGas

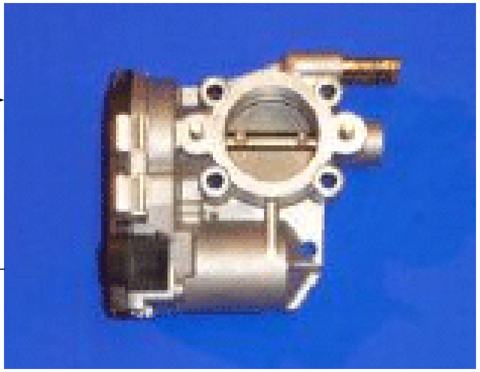


Throttle pedal
with position sensor

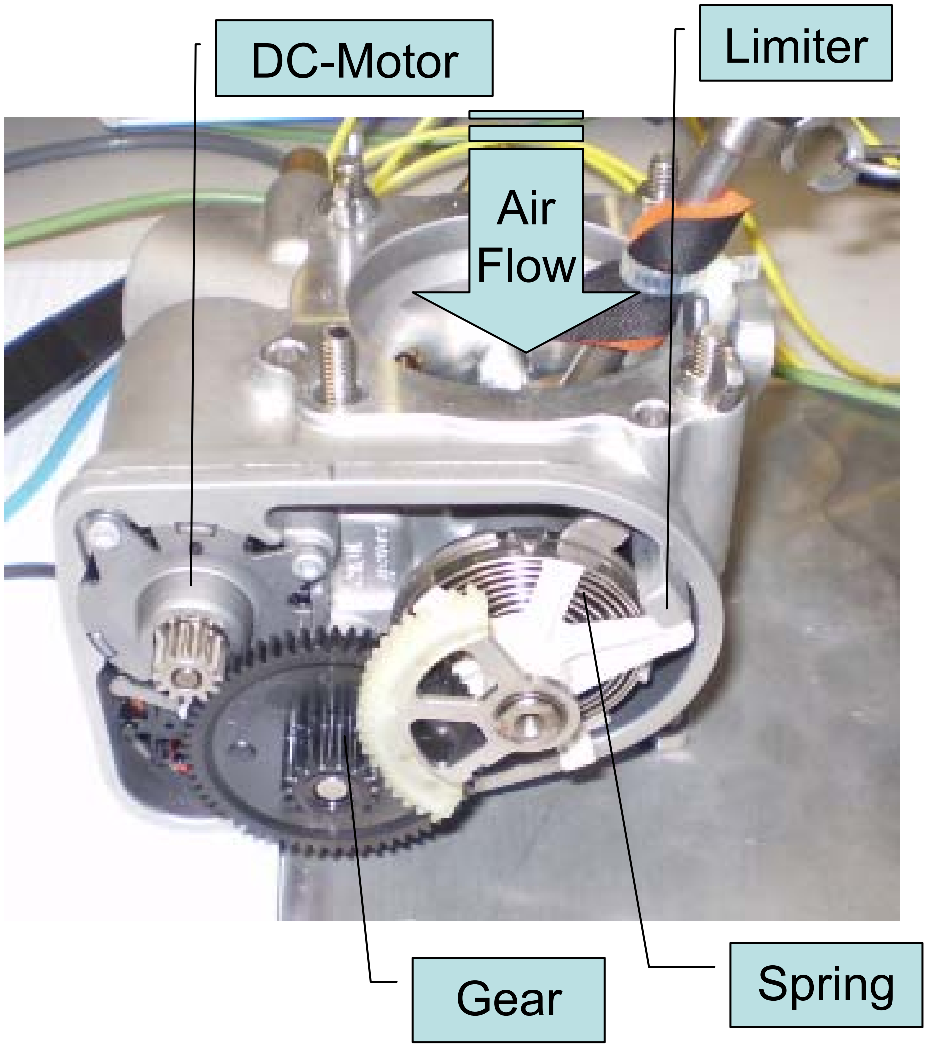
Controlling unit with
integrated logic and
power bridge

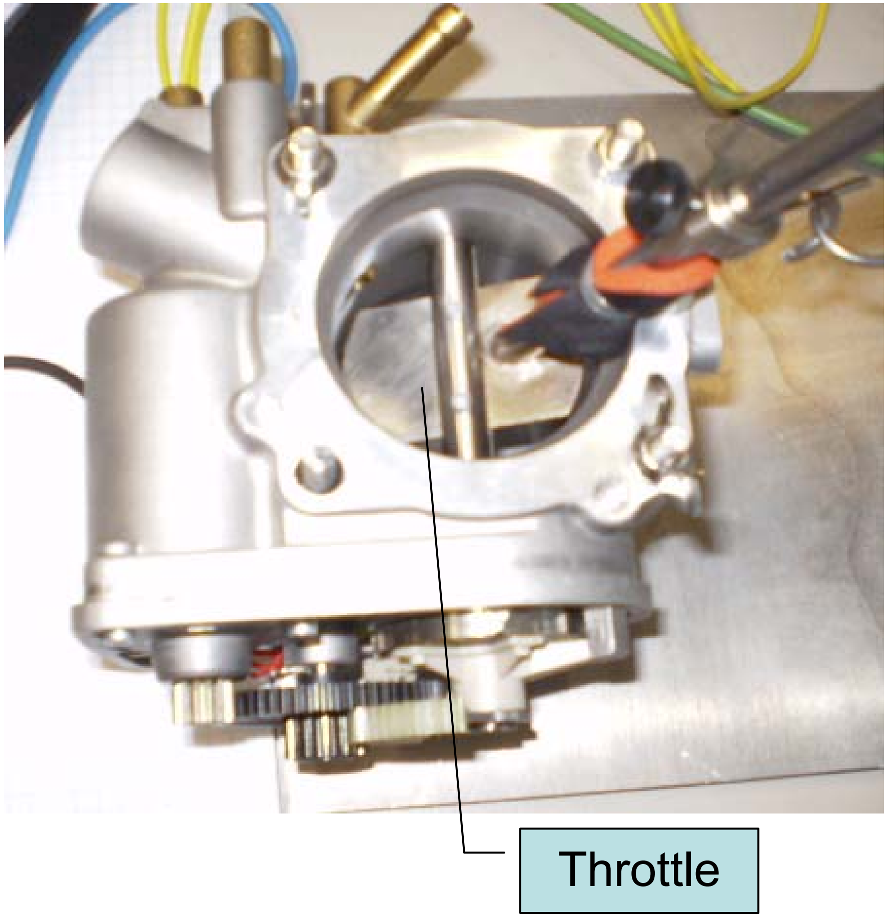
Electronic throttle
body



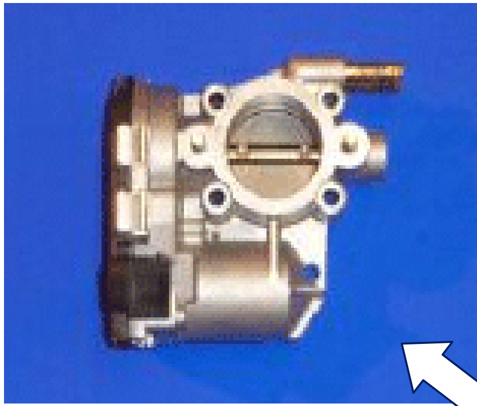


Throttle body



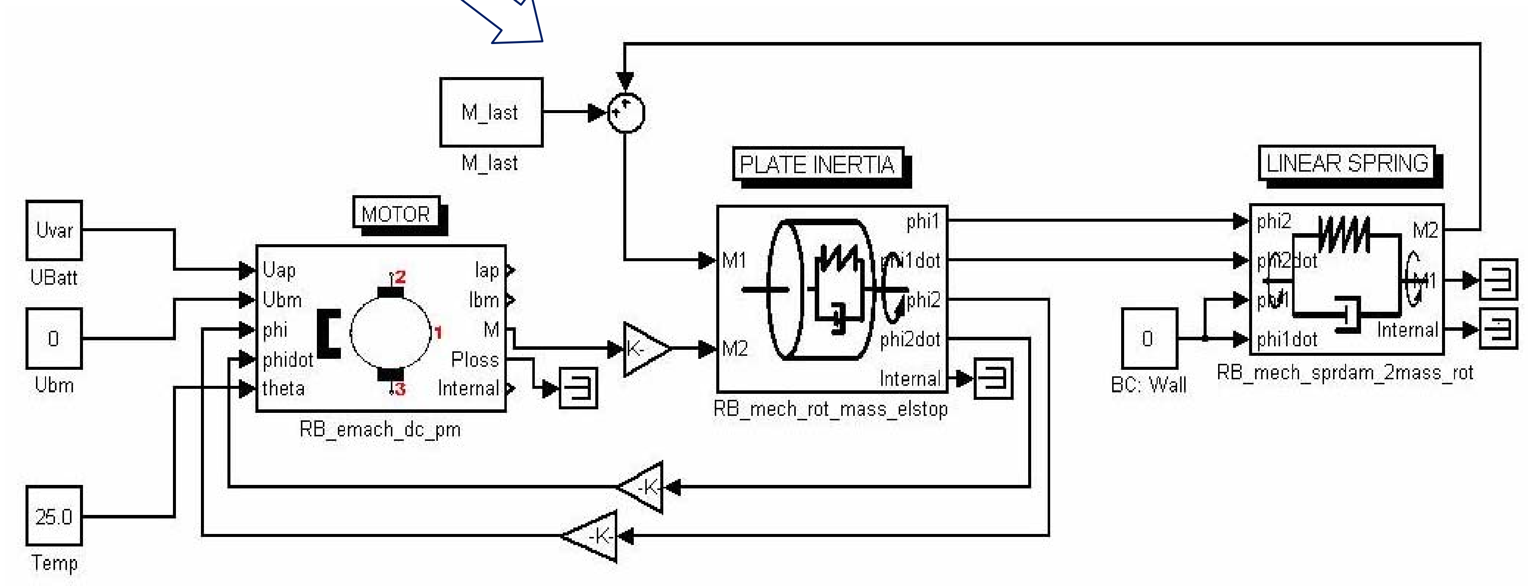


Throttle body model (Matlab Simulink)

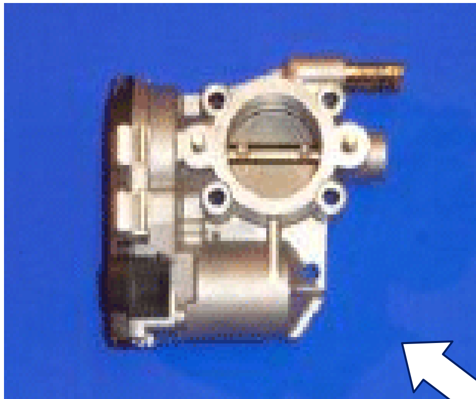


Matlab Simulink:

- Graphical programming (blockset)
- Fast results
- Avoid complex models and loops
- Only unidirectional signal lines

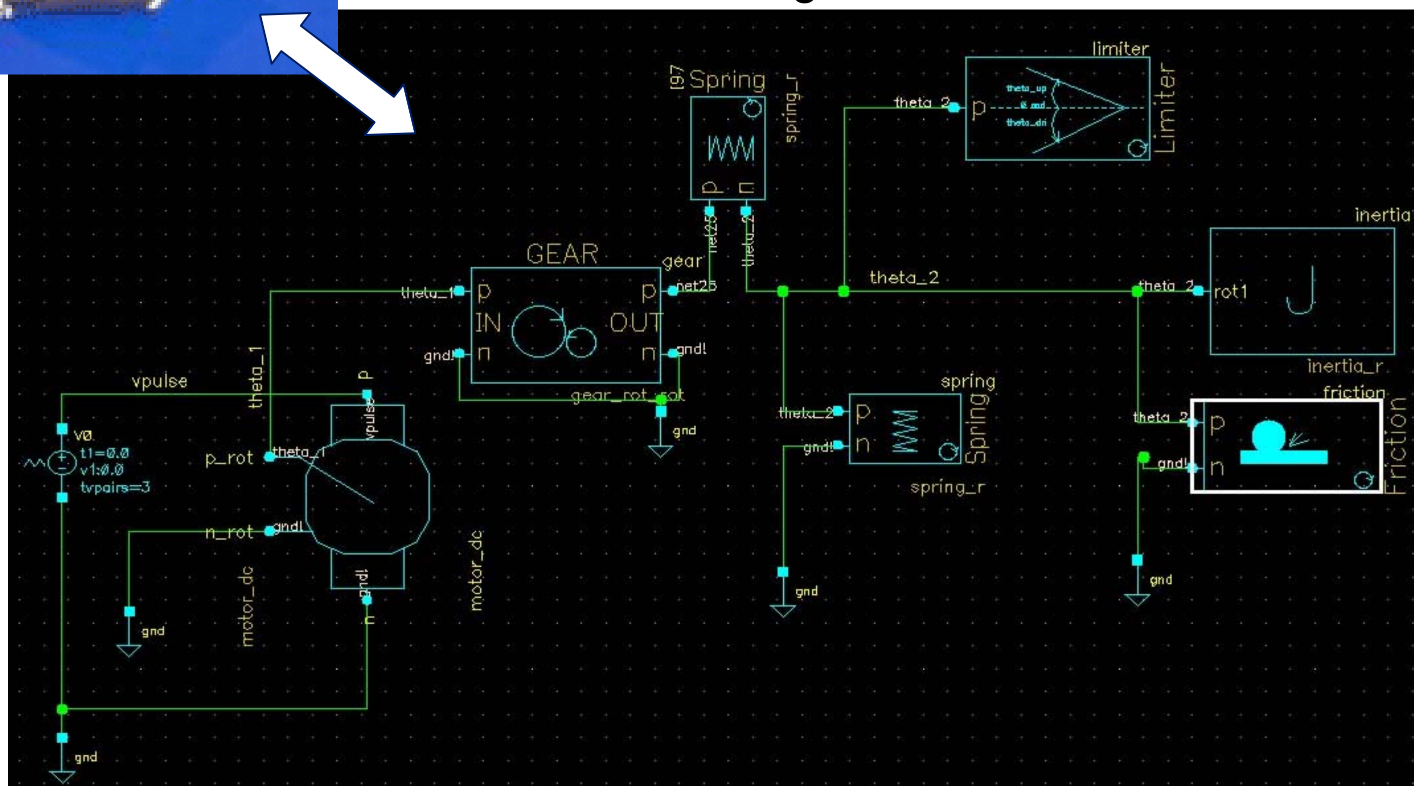


Throttle body model (VHDL-AMS)



VHDL-AMS

- VHDL knowledge prerequisite
- Complex models preferred
- Bidirectional signal lines



VHDL-AMS code example

Behavioral description of DC motor:
The DC motor transforms the electrical energy to mechanical energy.

Parameter declaration

Terminal declaration

```
entity motor_dc is

 generic (
  R_wind : real := 2.5;              -- Motor winding resistance [Ω]
  kt     : real := 0.025;            -- Torque coefficient [N*m/A]
  L_wind : real := 0.001;            -- Winding inductance [H]
  d      : real := 1.0e-6;           -- Damping coefficient [N*m/(rad/sec)]
  j      : real := 1.0e-6);          -- Moment of inertia [kg*m^2]

 port (
     TERMINAL p: ELECTRICAL;
     TERMINAL n: ELECTRICAL;
     TERMINAL p_rot: ROTATIONAL;
     TERMINAL n_rot: ROTATIONAL
    );
end entity motor_dc;
```

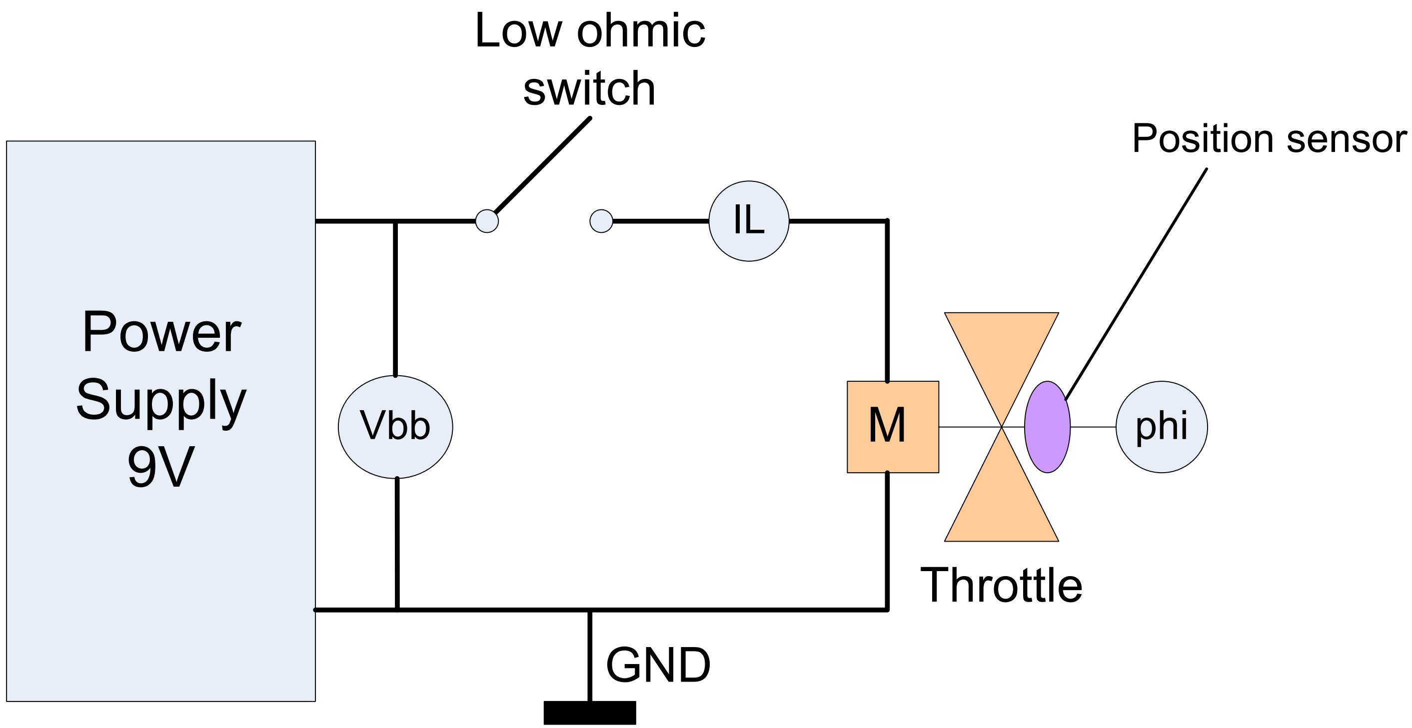
VHDL-AMS code example cont‘d

Across and through quantities

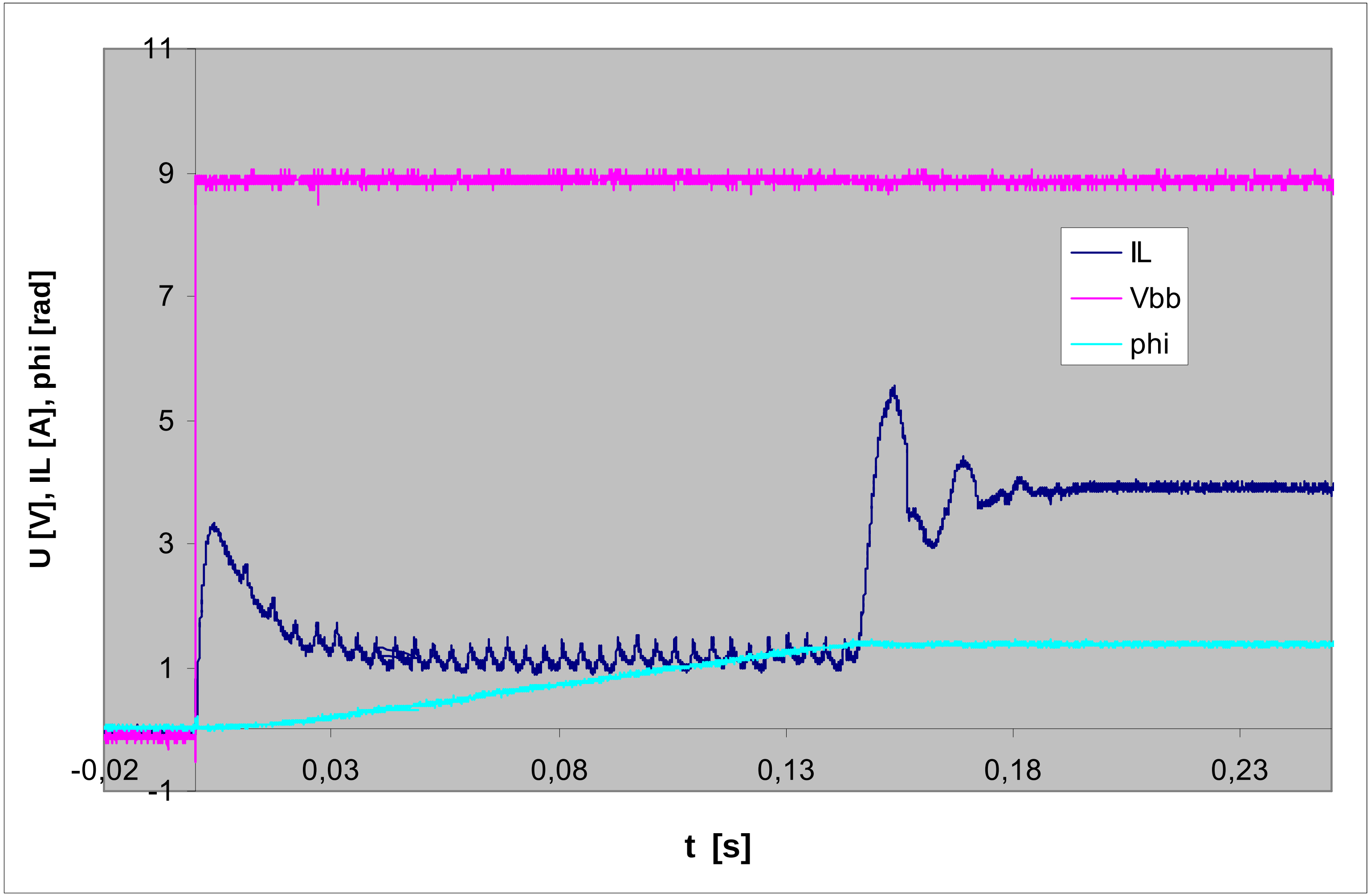
Simultaneous statements

```
architecture behav of motor_dc is

    quantity v across i through p to n;
    quantity theta across torque through p_rot to n_rot;
    quantity w : angular_velocity;

begin
    w == theta'dot;
    torque == -1.0*kt*i + d*w + j*w'dot;
    v  == kt*w + i*R_wind + L_wind*i'dot;

end architecture behav;
```

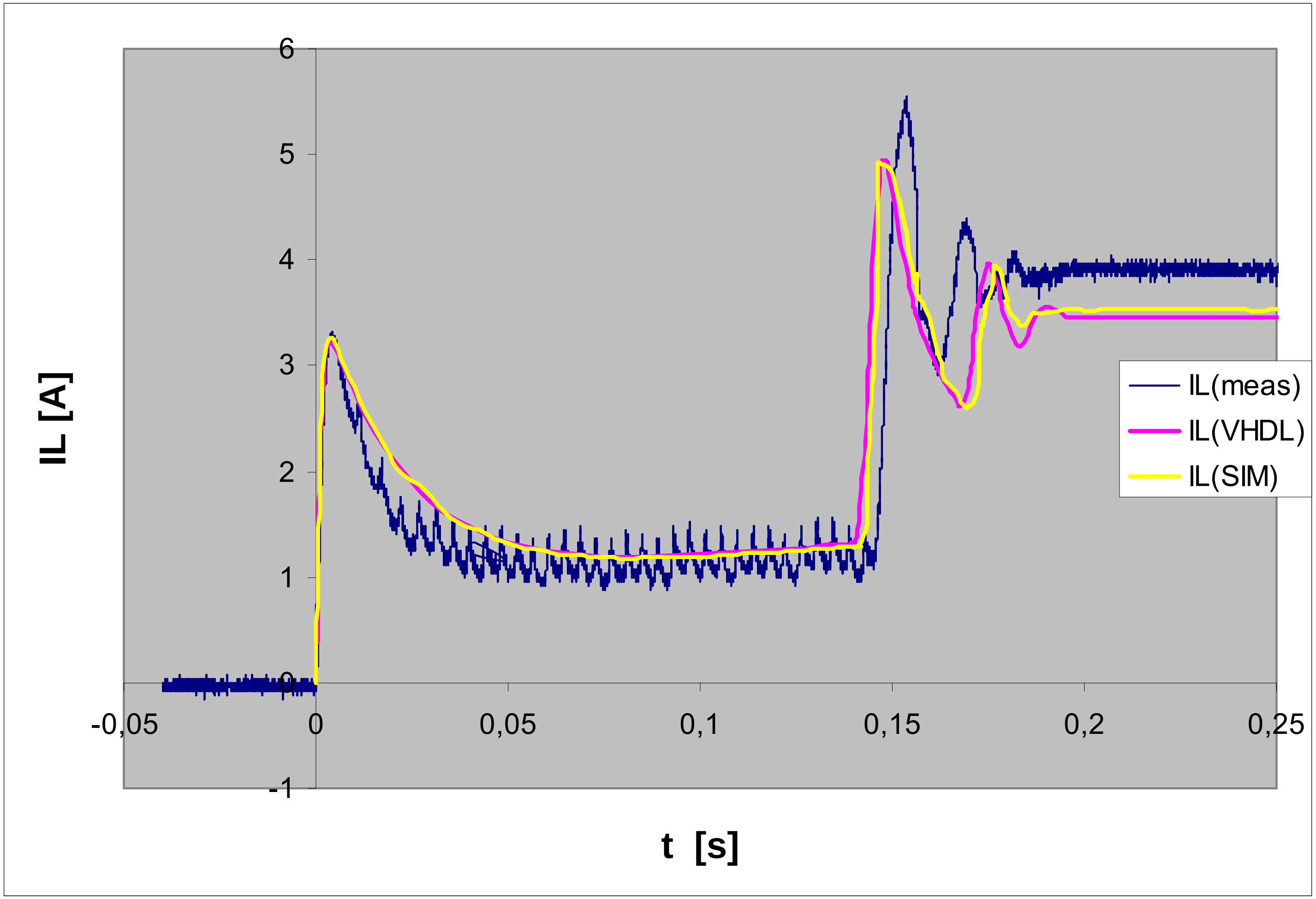
Measurement setup



Measurement results

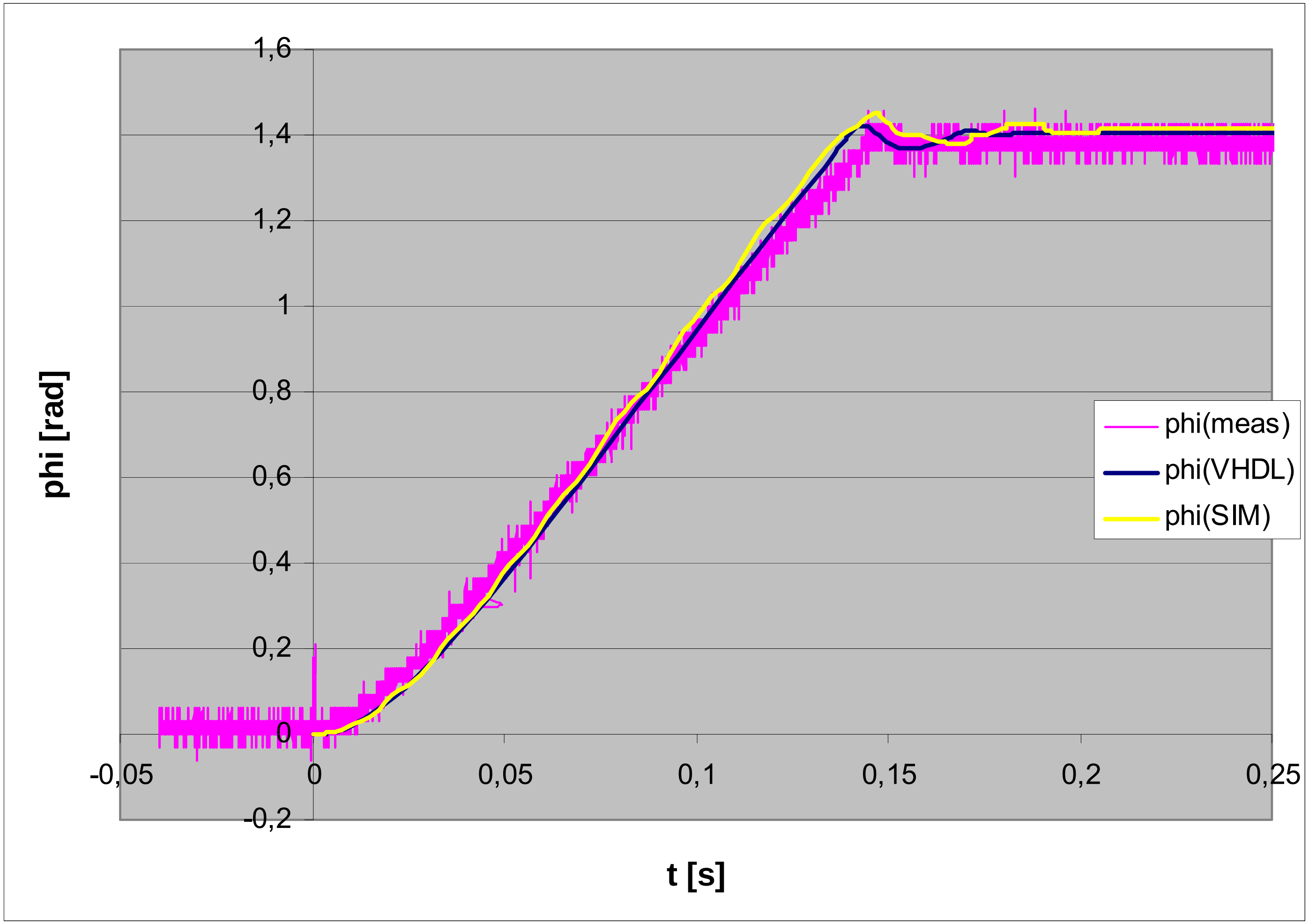


Comparing measurement with simulation results cont'd



Comparing measurement with simulation results

Throttle rotation: 80 deg, time: ~145ms



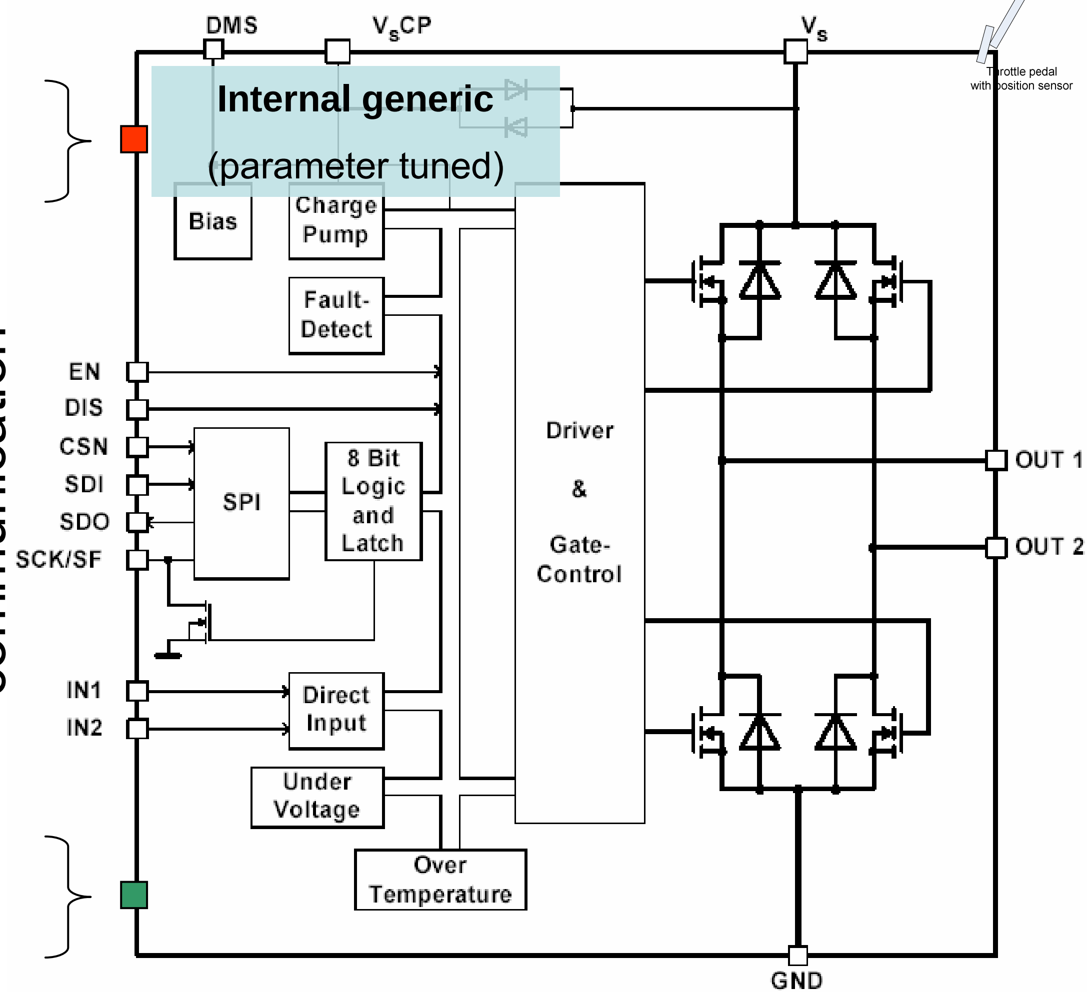
Controlling unit (1)

supply

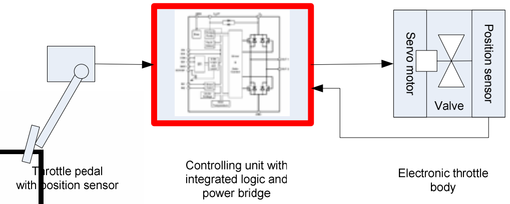
Tuning pin

(internal parameters)

communication



throttle



Thermal pin

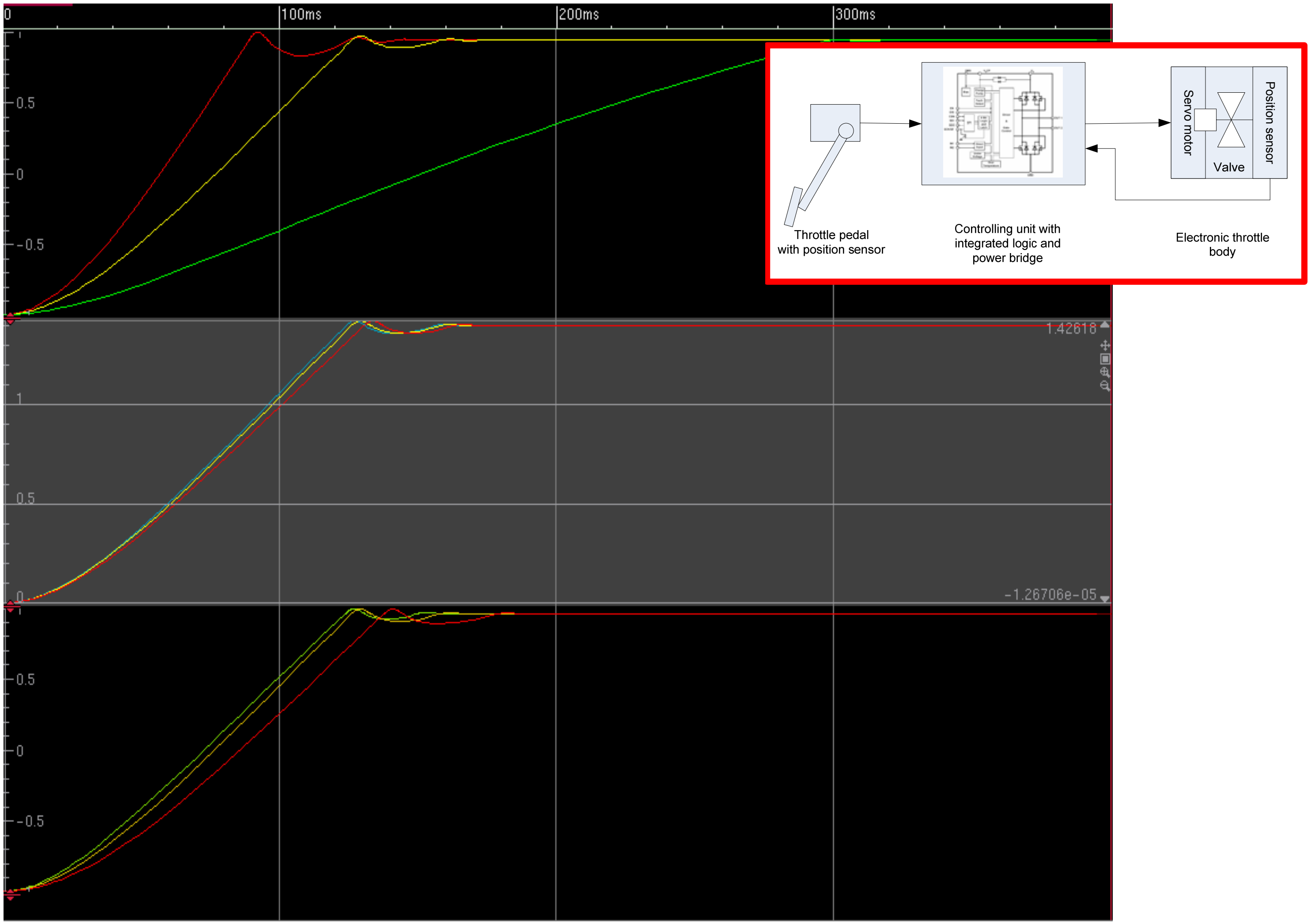
(external temperature)

Simulation of controlling unit with throttle

<u>Variation of:</u>
battery voltage:
18 V / 13 V / 8 V

temperature:
40°C / 25°C / 150°C

torque coefficient
kt+20% / kt / kt-20%



Thanks for your attention

Questions?

Controlling unit (2)

- In addition to the real chip pins, the VHDL-AMS model contains:
  - **1 thermal pin:** to be connected to an external thermal source, representing the ambient temperature
  - **1 electrical pin:** to be connected to an external electrical source, representing the image of the wished parameter variation
  - **1 internal generic:** to be set to the desired value, representing the parameter which is influenced by the tuning pin.

Worst-case modeling

Example:

generic value: Specifies the parameter to be varied

0 = Over-current limitation, 1 = RDSon value, …

electrical pin: Gives the amplitude of the parameter variation, in the range of the specification window

-1.0 = Minimum, 0.0 = Typical, 1.0 = Maximum