INSTITUT FRANCAIS
DES SCIENCES
ET TECHNOLOGIES
DES TRANSPORTS,
DE L'AMENAGEMENT
ET DES RESEAUX

Summer school HIL 2016
September 1&2, 2016

From simulation to real time control of an all electric bus : the ElLiSup project

B. Jeanneret, R. Trigui, D. Ndiaye
IFSTTAR – Site de Bron
Laboratoire Transports et Environnement

IFSTTAR

Outline

- **Brief presentation of the project**
- **Software In the Loop (Plant and Controller are both simulated)**
- **Processor In the Loop (Plant simulated/Controller on final processor) – Hardware In the Loop (Plant is part of real component/Controller on proc.)**
- **Rapid Control Prototyping (Plant is vehicle/Controller on proc.)**

ElLiSup project

A project supported by ADEME with the following partners:

ADEME – Agence de l'Environnement et de la Maîtrise de l'Energie · cea · EDF · ercteel – ELECTRONIQUE DE PUISSANCE EMBARQUEE · IFP Energies nouvelles · IFSTTAR · IVECO BUS · MICHELIN – Une meilleure façon d'avancer · RATP · RECUPYL

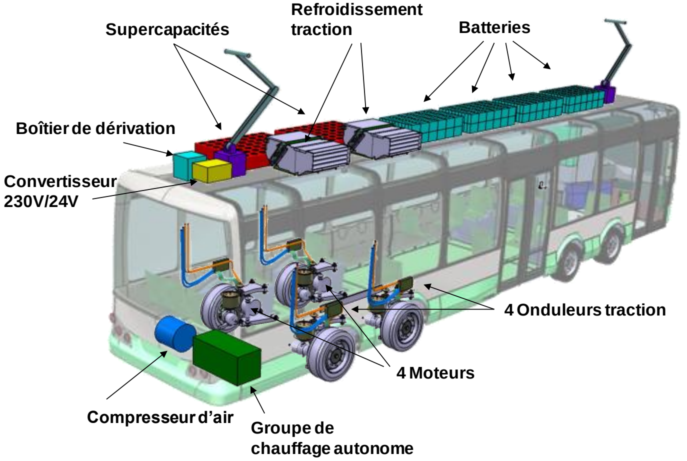
Project objectives

<u>Purpose</u> : Electromobility for public transport.

- Series hybrid bus 12m long - 3 km ZEV
- All electric bus with a dual energy storage system composed of batteries and supercapacitors 12m long – 8 km ZEV

Fast charging system with catenary at the end of the line (up to 250 kW)

Véhicule tout électrique avec recharge rapide en fin de ligne



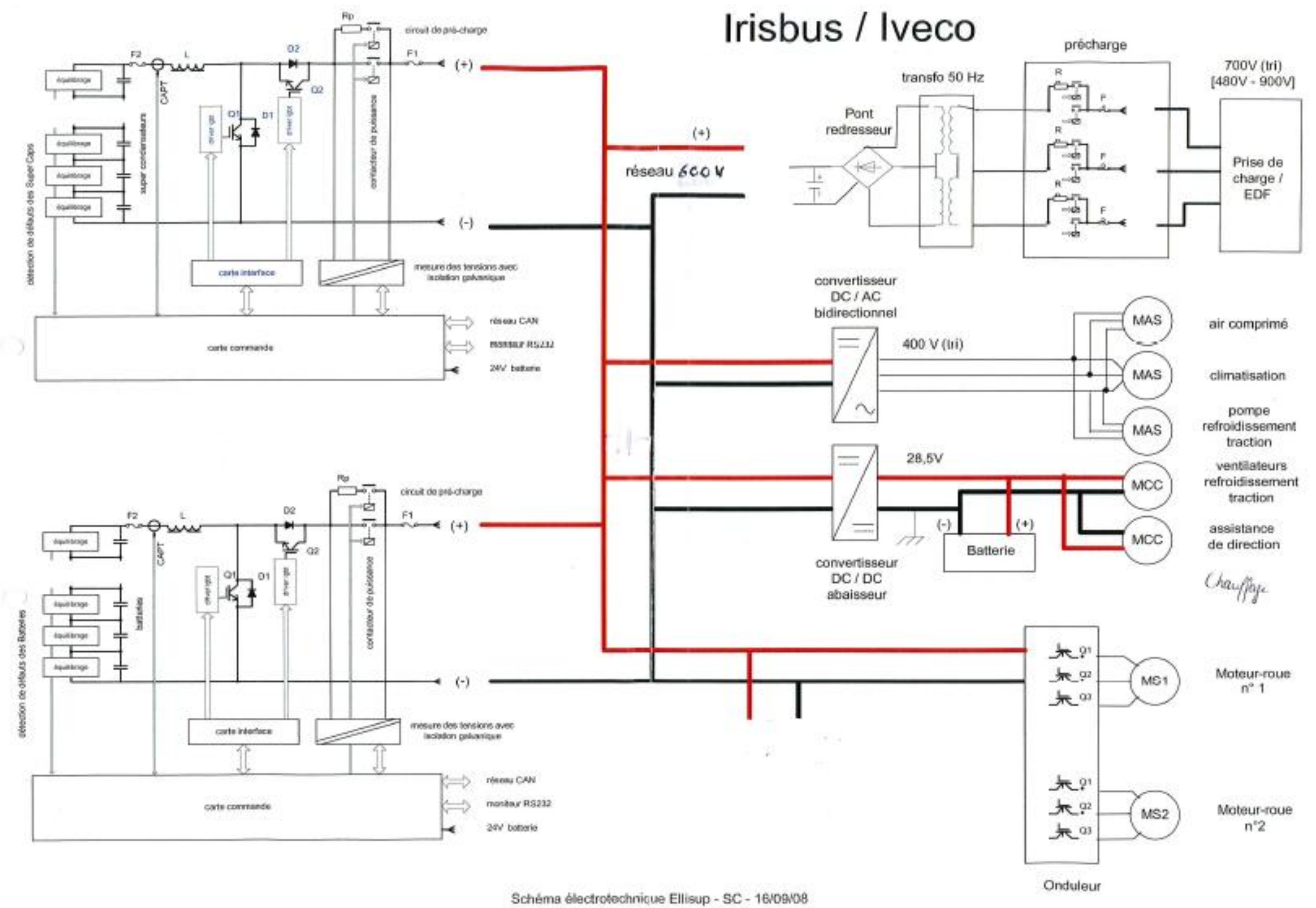
IFSTTAR contributions:

- Battery caracterization & selection due to this specific usage
- Modeling and energy management development of the dual system (batteries & supercapacitors)
- Realization of the prototype supervisor

The vehicle

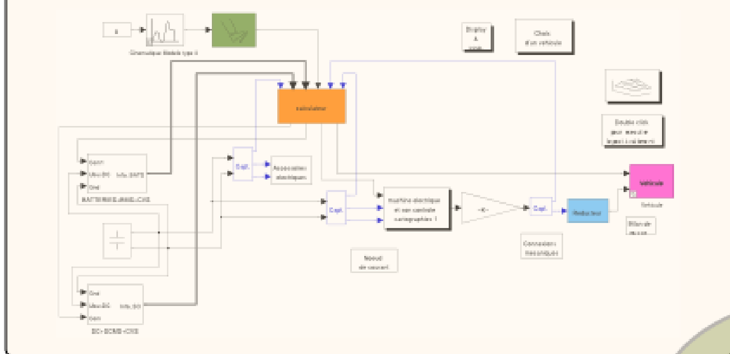
- A bus with 4 axles : 3 steering axles, 2 driven axles
- Small wheels : 17 inches (small diameter for increased interior space)
- 4 electric motors of 50 kW each located in the wheel
- 4 batteries packs of 80 kW each
- 1 supercapacitor pack of 80 kW
- DC/AC convertors (380V et 24 V) for vehicle's auxiliairies (power steering, air compressor, fans…)
- A fast charging system with catenary

Electrical Architecture

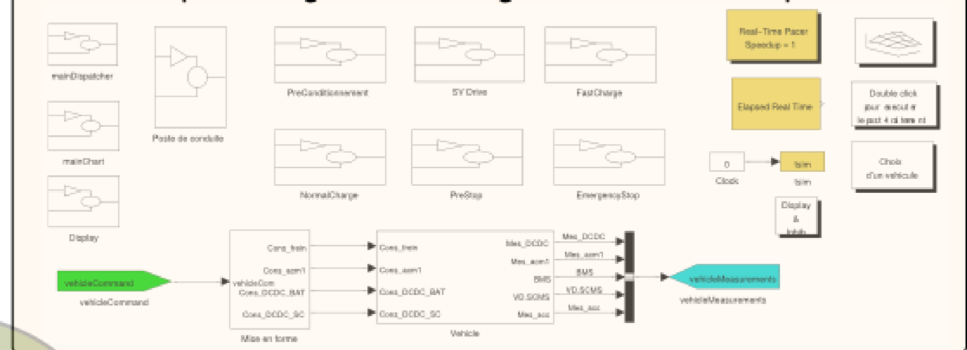


Main development steps for the supervisor

Step 1: off-line simulation

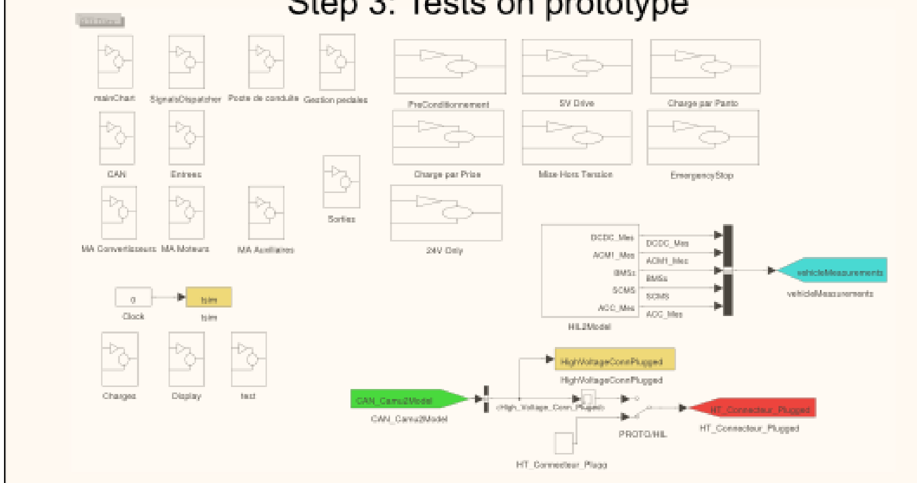


Step 2: Progressive integration of real components



Vehicle supervisor

Step 3: Tests on prototype



Step 1: Model in the loop (MIL)

- Objective : energy sharing between battery and supercapacitor
- Backward models (from the wheels to the energy sources) are used to find optimal solutions regarding objective functions
  - A priori knowledge of the vehicle mission
  - Dynamic programming, Pontryaguin minimum principle
- Forward models are developped to find sub-optimal solutions applicable in real time

Backward Models - MIL1

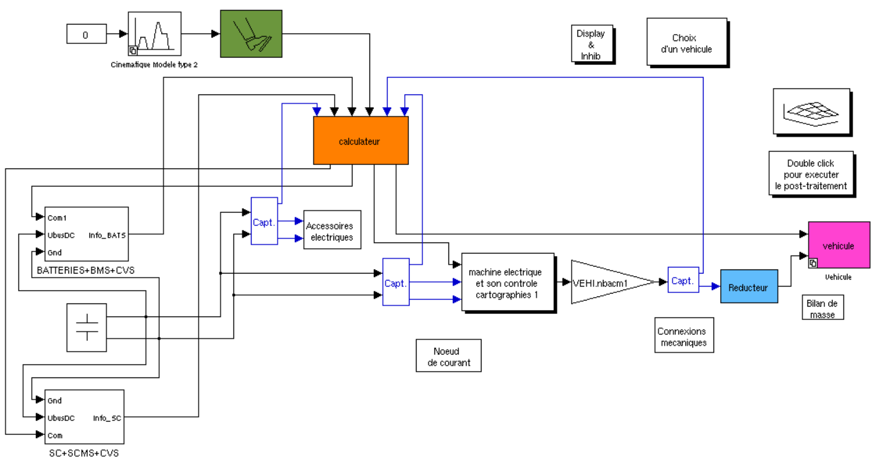
- Simplified model
- Global Optimization
- A priori knowledge of vehicle mission

Backward Models - MIL2

- Sub-optimal strategy
- No knowledge of vehicle mission

Forward Models - MIL3

-More realistic model
- Sub-optimal strategy
- No knowledge of vehicle mission
- Introduction of the driver in the loop
- Controler ready for SIL introduction



Examples of solution studied in this step

- 3 levels for control and energy management strategy
  - Level 1 :
    - SOC regulation : Power demand function of SOC of each branch

    CVS power: $P_{CVS_i} = P_{res} * \left(1 - (soc_i - \overline{soc})\right) / 4$

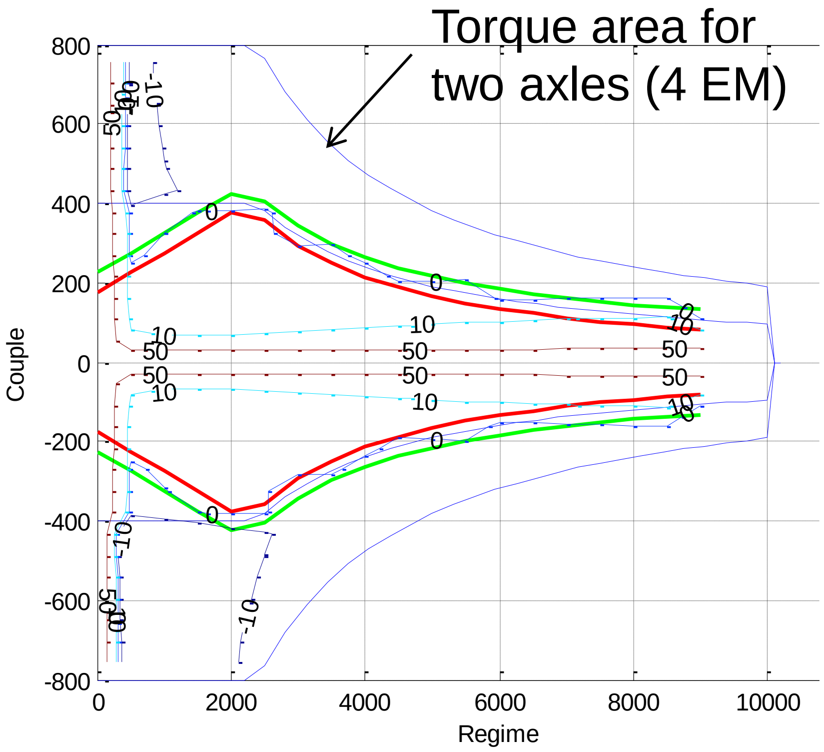
  - Level 2 :
    - Loss minimization by adapting voltage level as a function of vehicle speed
    - Selection of active axle

  - Level 3 :
    - Sharing power between battery and supercapacitor
      - Static look up tables (default strategy)
      - Dynamic control in order to minimize battery RMS currrent

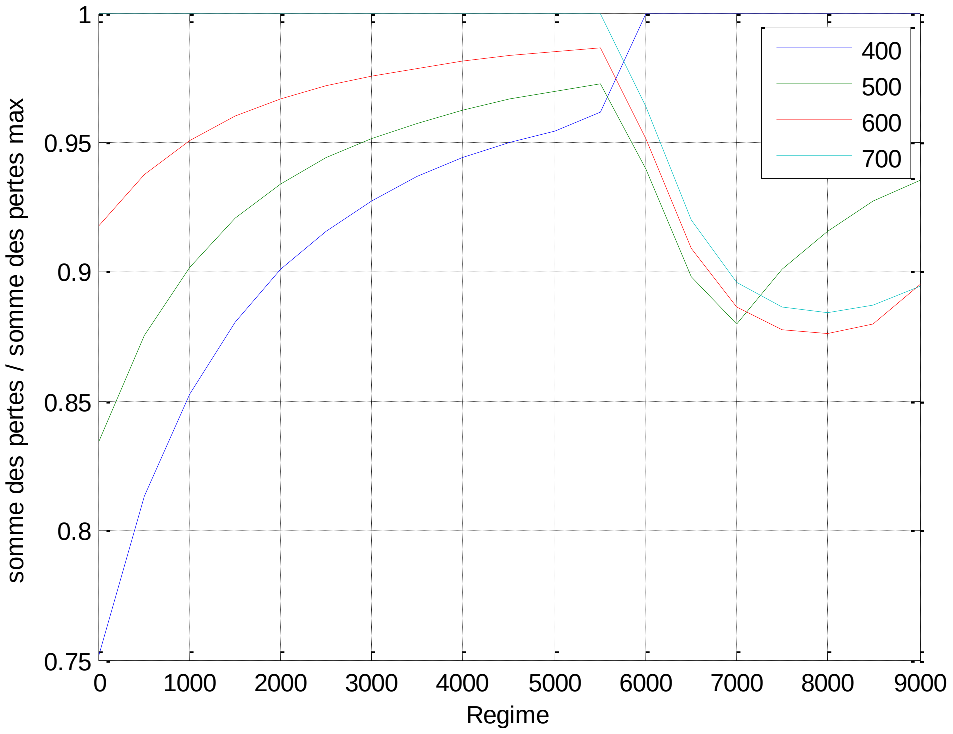
$$\text{Minimize} \quad J = \int_{t_0}^{t_f} I_{batt}^2 \, dt$$

Example of optimization of level 2

Mapping of gains/losses between one and two axles

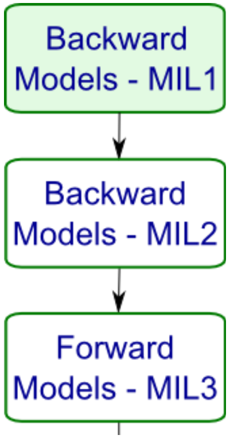


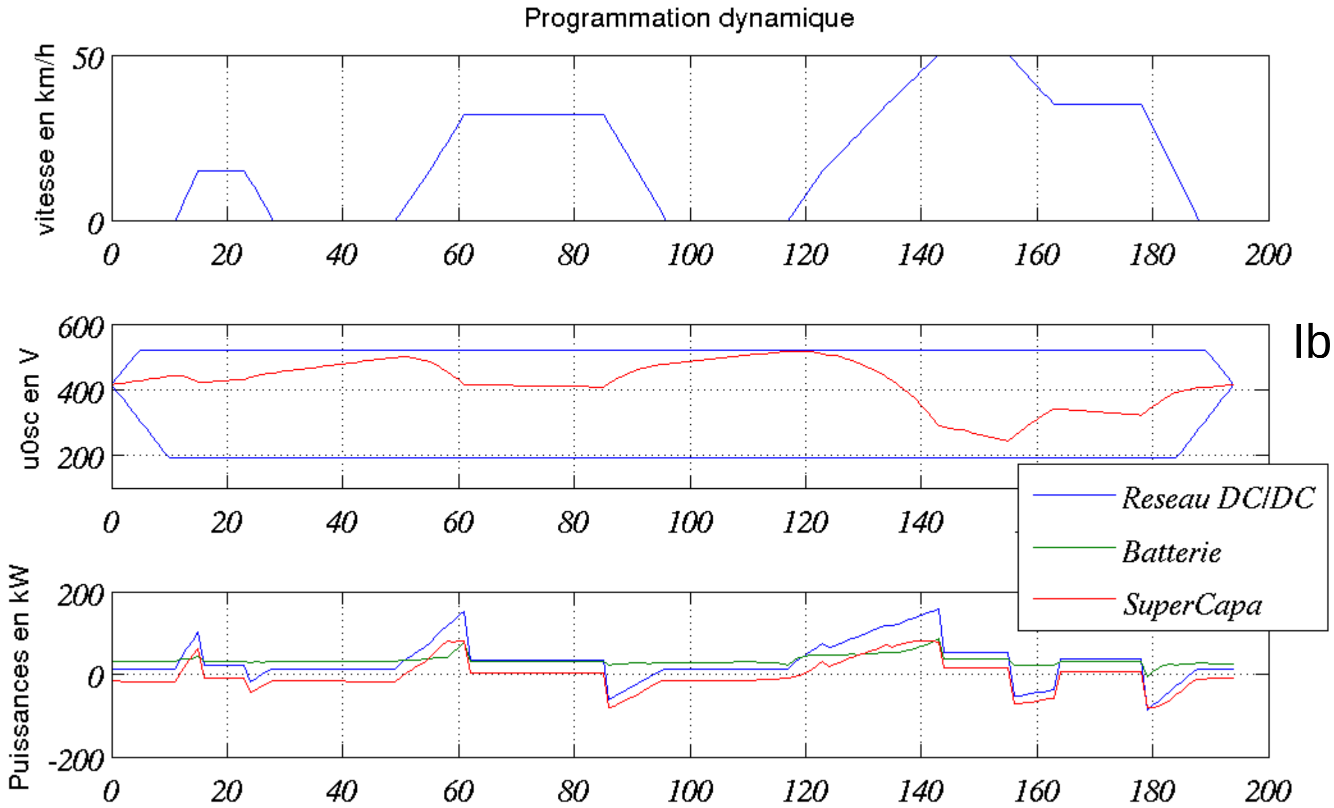
Minimizing losses by adapting DC bus voltage as a function of speed



Both strategy can be cumulated: This leads to a reduction of 2 to 4% of battery energy depending on the cycle

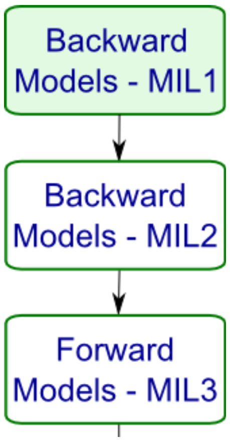
Example of level 3 optimization
MIL1 : backward - dynamic programming

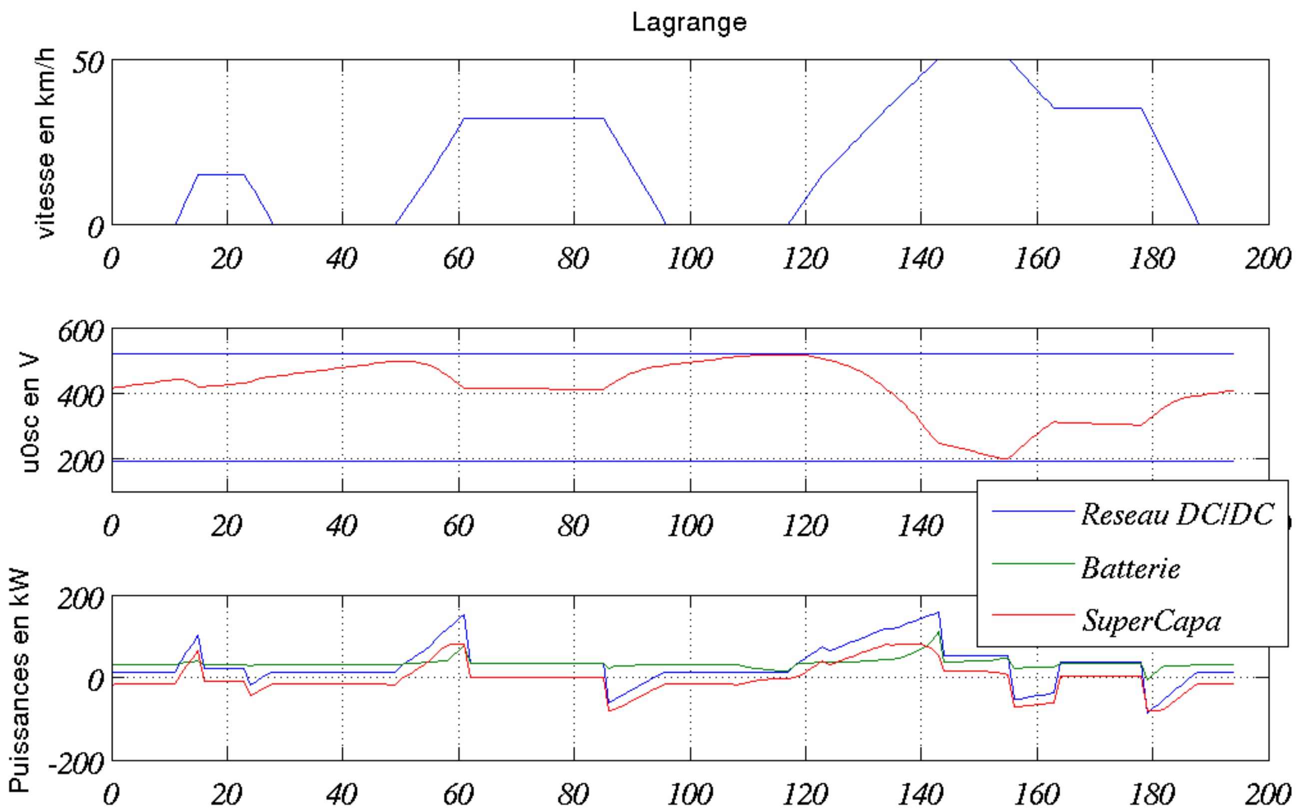




$Ibat_{RMS}$=75,1 A

Example of level 3 optimization
MIL1 : backward - Pontryaguin Minimum Principle



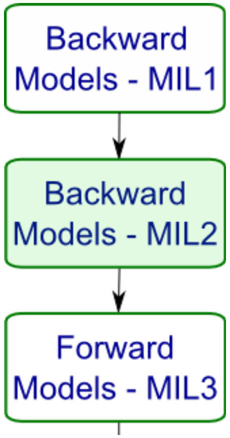


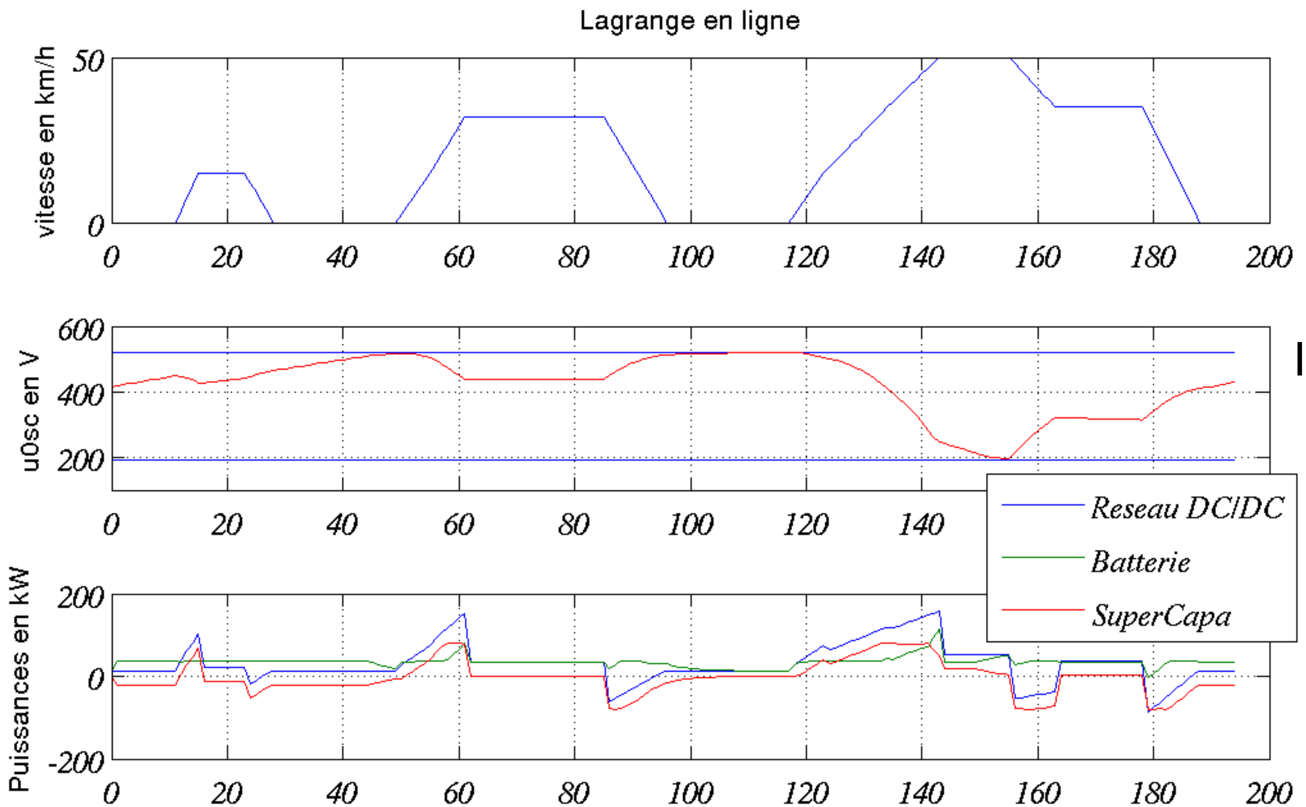
$Ibat_{RMS}$=75,7 A

Minimize at each time step :

$$H(E_{uc}, P_{uc}) = I_{batt}^{2}(P_{uc}) + p \cdot P_{0\,uc}(E_{uc}, P_{uc})$$

Example of level 3 optimization
MIL2 : backward - Simplified calculation based on PMP



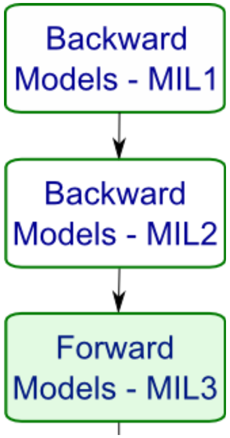


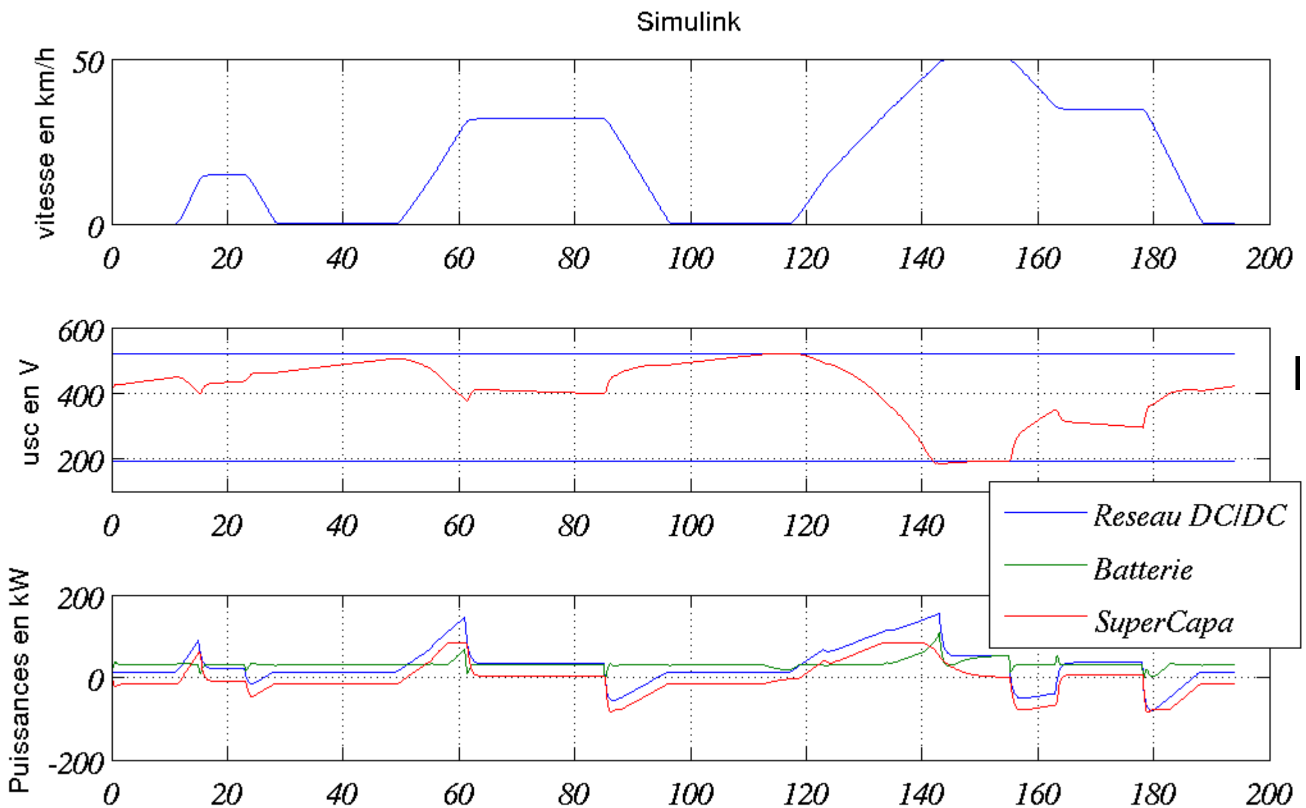
$Ibat_{RMS}$=78,2 A

UC power :

$$P_{uc} = \left(2 \cdot \frac{P_{res}}{U_{bat}^{2}} - p \right) \cdot \frac{U_{bat}^{2}}{2}$$

Example of level 3 optimization
MIL3 : forward model – Simplified calculation based on PMP





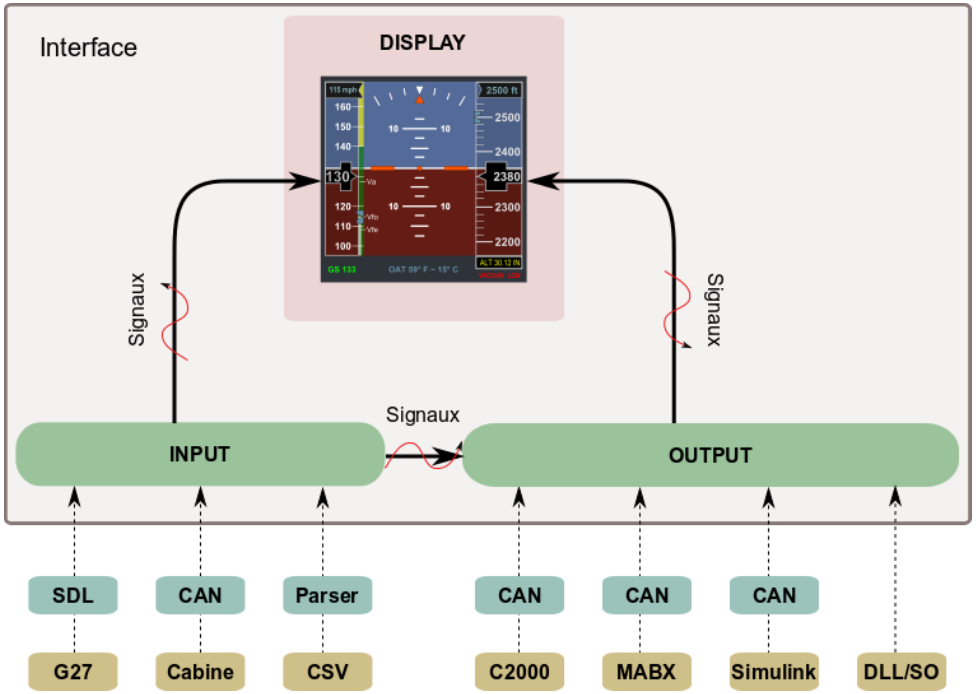
$Ibat_{RMS}$=74,8 A

- Choose an initial value for lagrange parameter, p
- Add a regulator to stabilize UC level of energy

Step 2 : Progressive integration of components

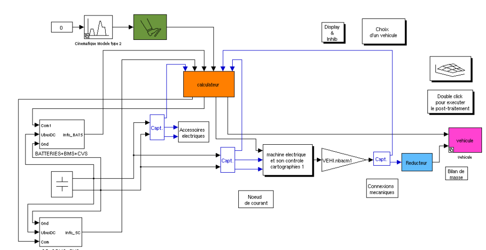
- A transition between Processor in the loop (PIL) to Hardware in the loop (HIL)
- At the beginning of this step, the model can even be compiled in the hardware
- The real components are progressively suppressed from the simulation model and integrated in the project
- An intensive use of test bench
- 2 examples:
  - Step 2.1 : Integration of the driver in the loop
  - Step 2.2 : Testing the application in an engine test bench

Step 2.1 : Driver in the loop test



Development of a framework (MODYVES project) to connect any kind of input (driver input) to any kind of output (vehicle model)

- Python code
- Application running on windows
- Windows timers
- « Soft » real time application
- Use of SDL library (G27)
- Peak or Systec usb adapter



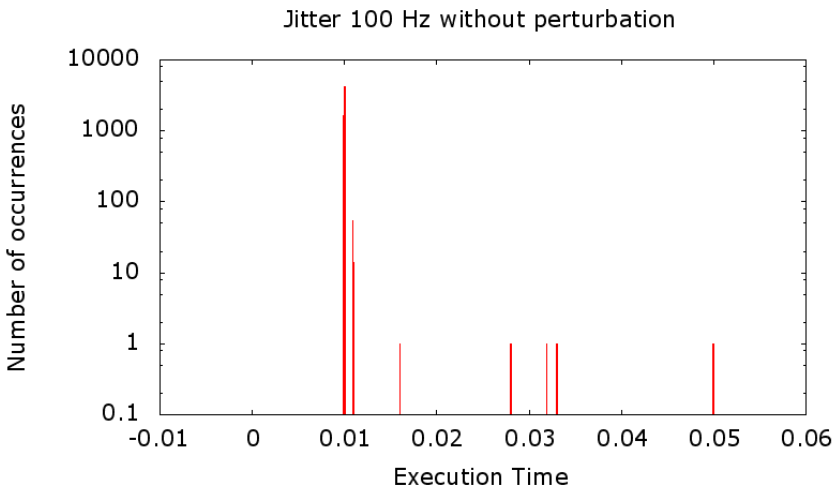
Functions tested:

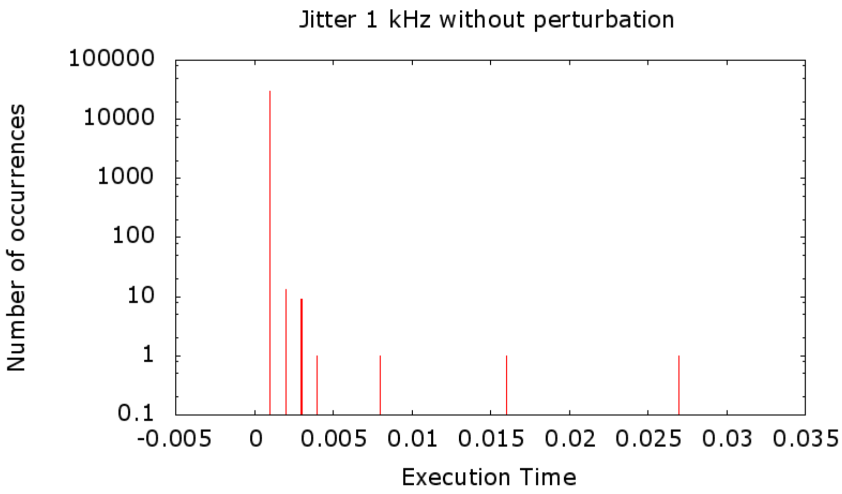
- Forward and reverse speed
- Recovery braking modes
- Anti move back

A parenthesis : jitter response for this « soft » RT Modyves framework

Jitter response for the Modyves framework and two theoretical period of 100Hz and 1 kHz (~1 mn)

- Intel Core i7 3610QM 2.3 GHz
- Windows Seven





Mean dt = 0,01003
Max = 0,053
Min = 0,00999

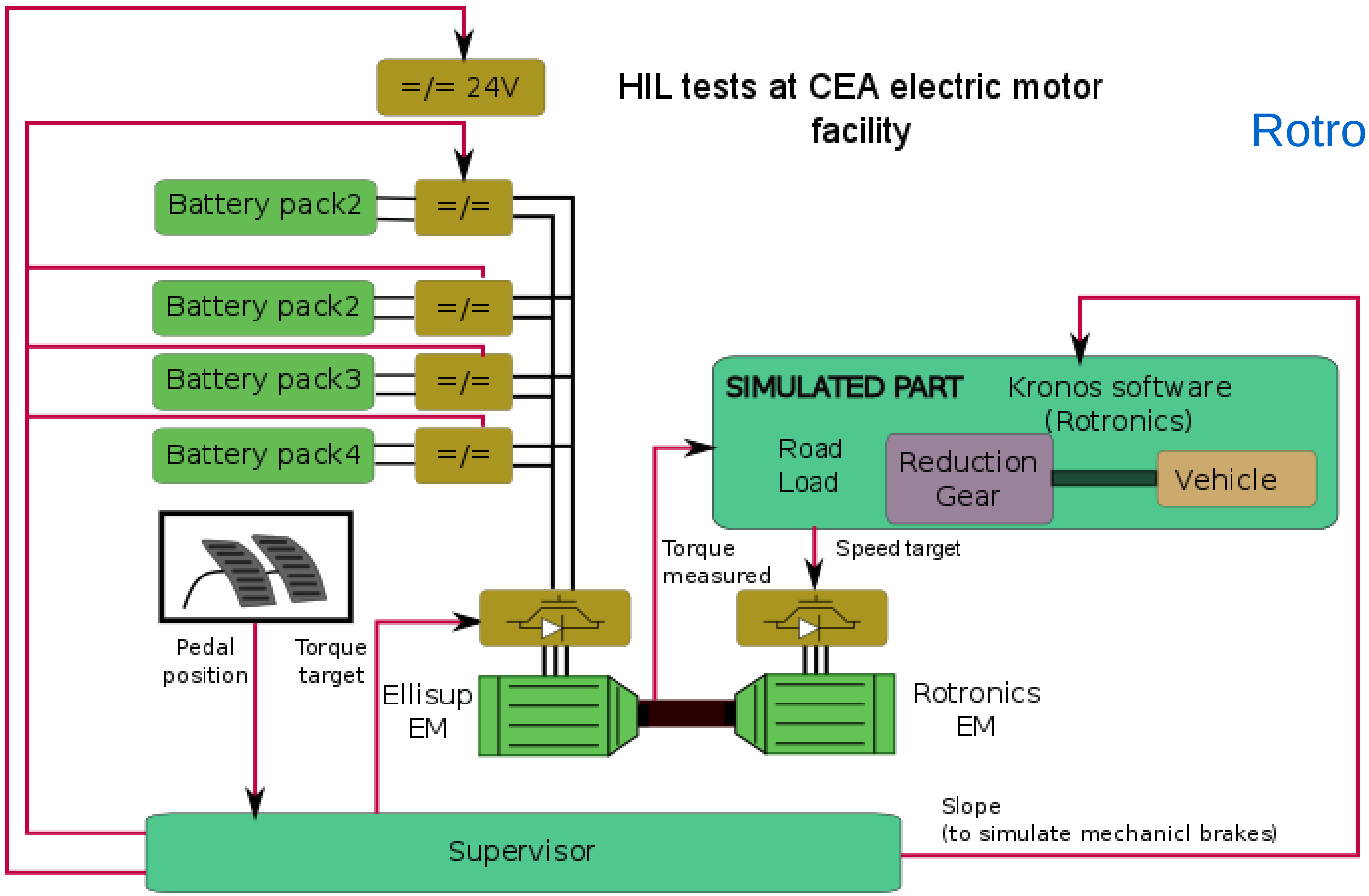
Mean dt = 0,001002
Max = 0,0027
Min = 0,0009999

According to the pc characteristics, deviation from theoretical frequency could be important, but still far from human time response

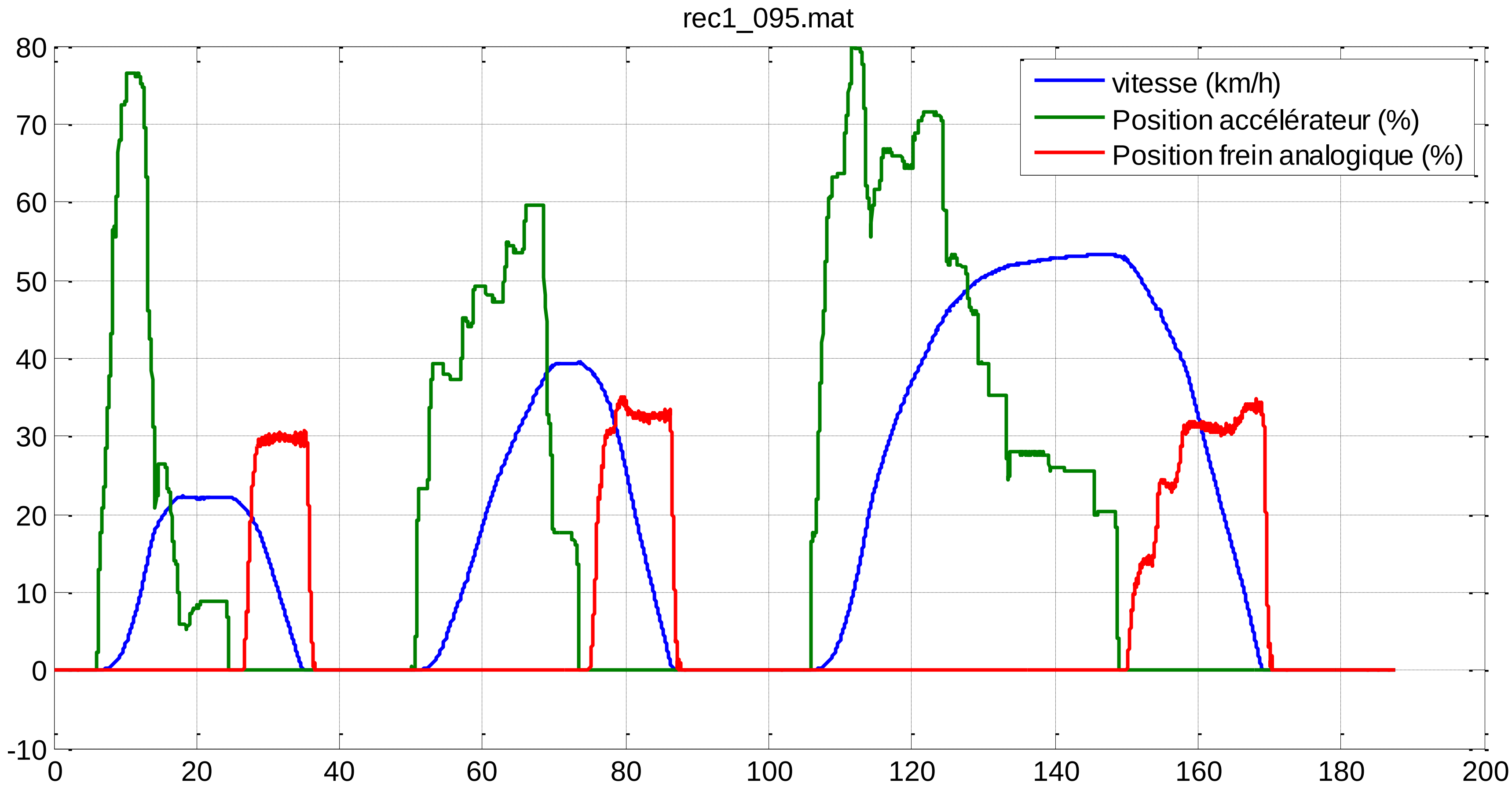
Step 2.2 : HIL test on engine test bench

HIL tests at CEA electric motor facility

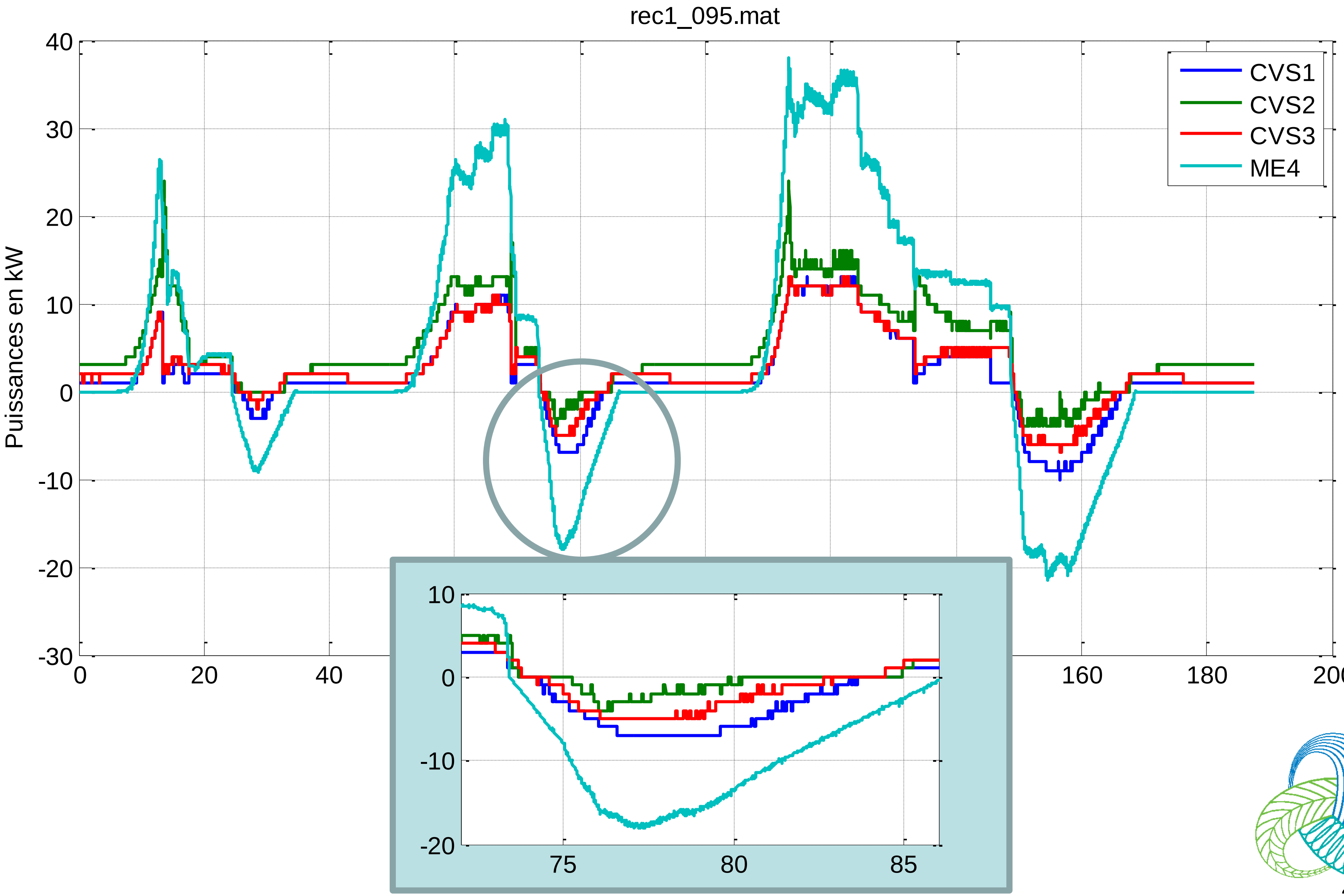
Rotronics bench



Step 2.2 : HIL test on engine test bench



Step 2.2 : HIL test on engine test bench

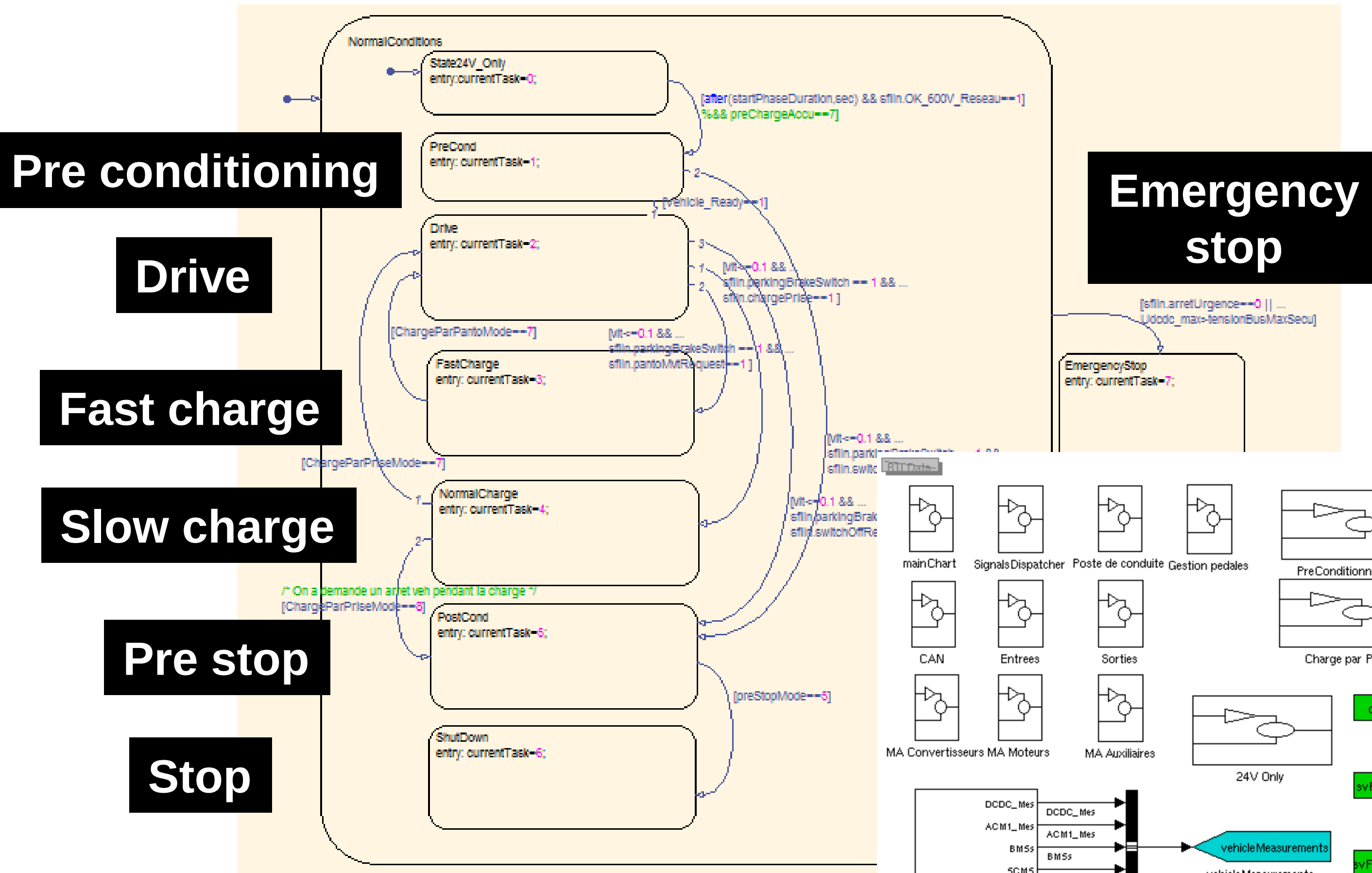


Step 3 : Control Prototyping with the final supervisor

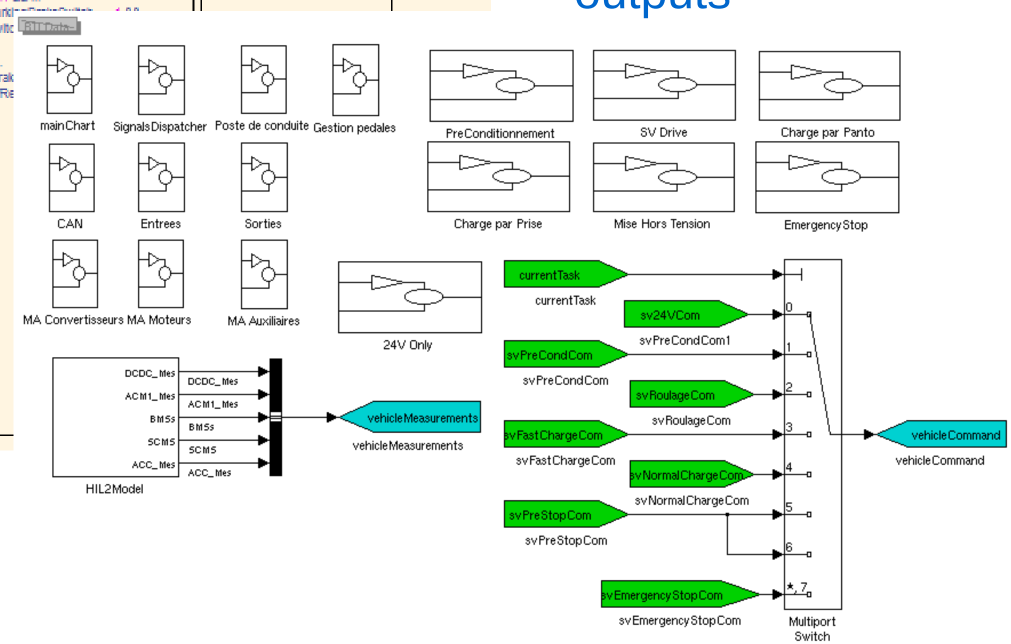
- Coded in Simulink (~6000 elementary simulink blocks) with a many Stateflow charts on a dSpace micro-autobox
- Single tasking/single rate, loop frequency =1 kHz
- Four CAN network (Vehicle, EM, BMS and DC/DC converter, auxiliaries). For each critical frame, Rx time is scheduled in order to detect a default in the communication between ECU.
- ~20 analog or digitial inputs/outputs
- Wired Safety Lines between the supervisor and the electric machines, in redondance with a CAN based safety Line.

Structure of the supervisor

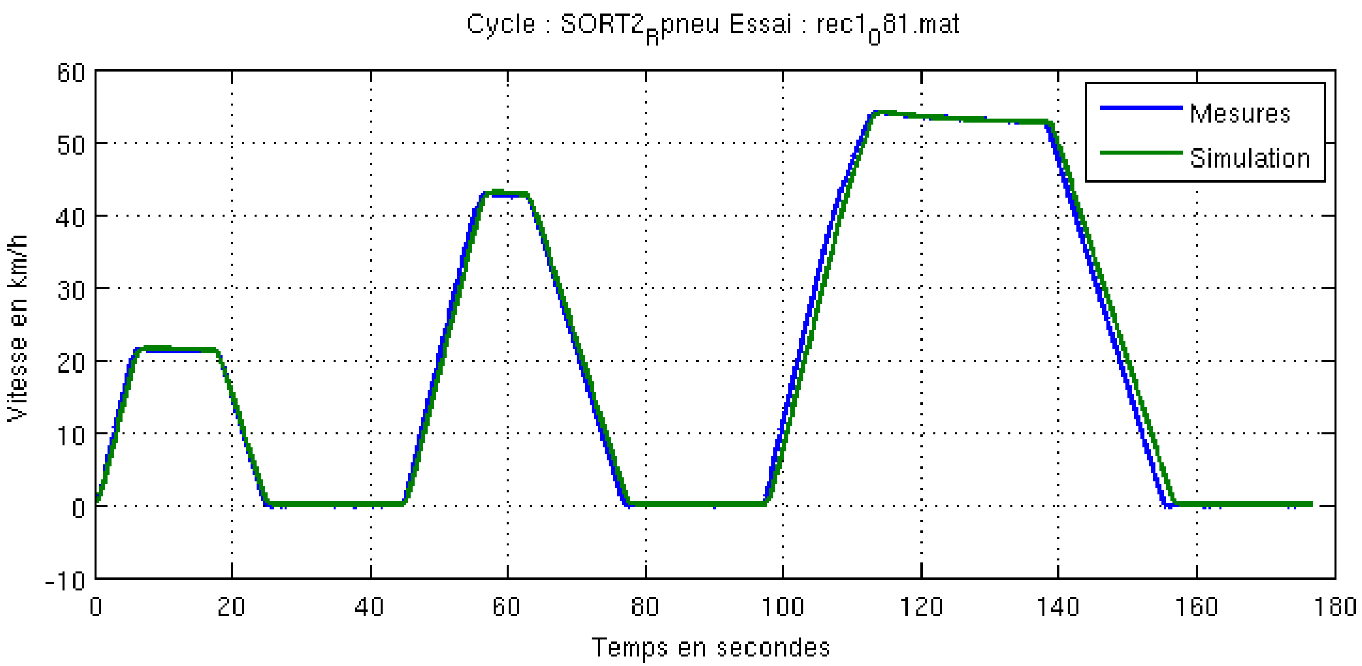
Each state of the diagram is associated with meta blocks which outputs the appropriate command



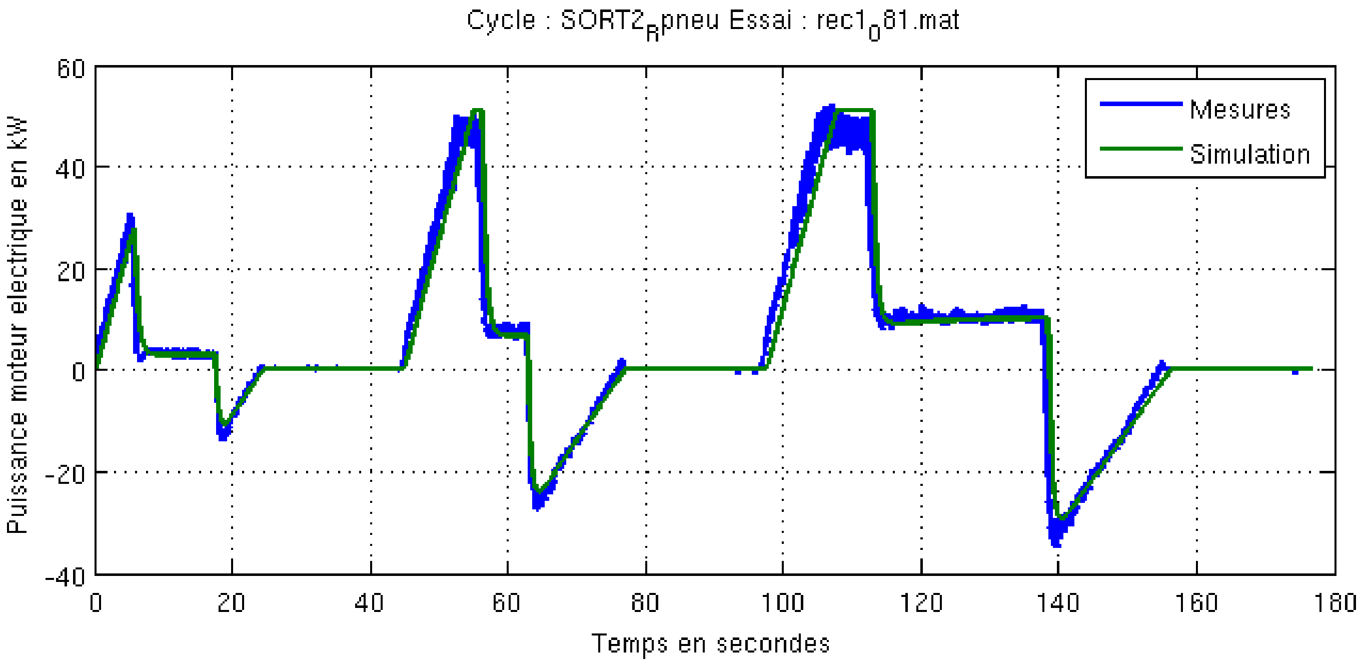
Two main data bus are consolidated for inputs and outputs



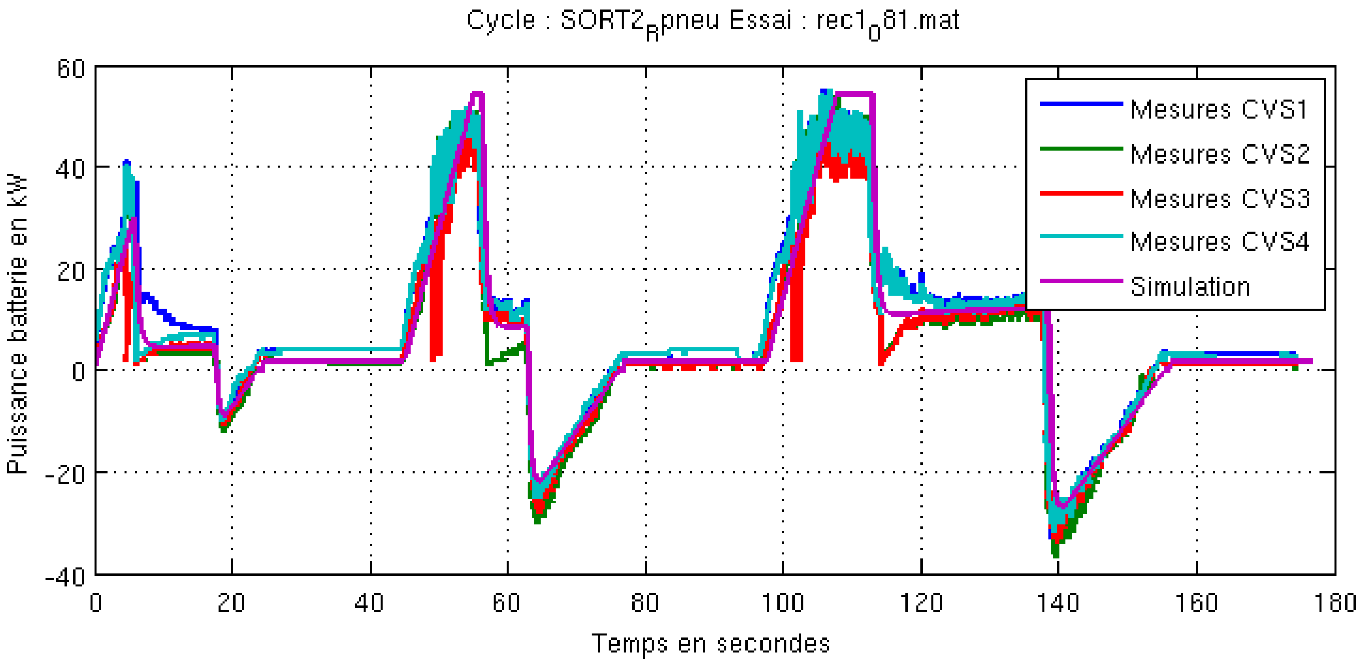
Comparisons between measure and simulation on SORT2 cycle



Electrical power of one motor



DC/DC converters power



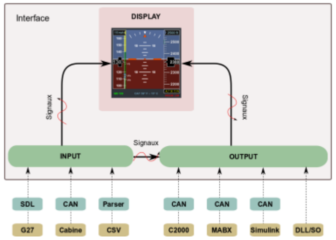
- Some difficulties to stabilize the different converters power
- Each DC/DC ECU has its own low level control

Conclusion

- Electric bus with complex architecture has been designed
- Different levels of control were studied
- A progressive methodology of controller design is adopted :
  - Simulation approach (from simple to more realistic models)
  - Processor in the loop
  - Hardware in the loop
- This approach allows to built optimal control for energy management and supervisor
- Prototyping hardware makes the debugging phase more easy, but it’s not an industrial solution

- C2000 cards from TI have been successfully tested with simulink applications and adapted to our needs
  (2 CAN, 16 ADC, 16 DI, 5 DIO, 4 PWM, 2 DAC)

- Modyves framework wants to be as generic as possible in order to connect any kind of inputs (example: the driver) to any kind of outputs.



Thanks for your attention

Ifsttar

Contacts :

Bruno.jeanneret@ifsttar.fr

www.ifsttar.fr