VPPC 2014
IEEE Vehicle Power and Propulsion Conference
"Spreading E-Mobility Everywhere"
October 27-30, 2014 — Coimbra, Portugal http://www.vppc2014.org

MEGEVH
French network on HEV's

Optimal control of hybrid powertrain with the Pontryagin Minimum Principle

Part 2 – Real time control

S. Delprat

University of Valenciennes et du Hainaut Cambrésis

Summary

Part 2 : Real time control

I) Introduction

II) Regression based algorithms

III) ECMS Family

- Basic algorithm
- Predictive control scheme

Acknowledgments

This work has been supported by:

European Funds F.E.D.E.R.

The French Nord Pas-de-Calais Region

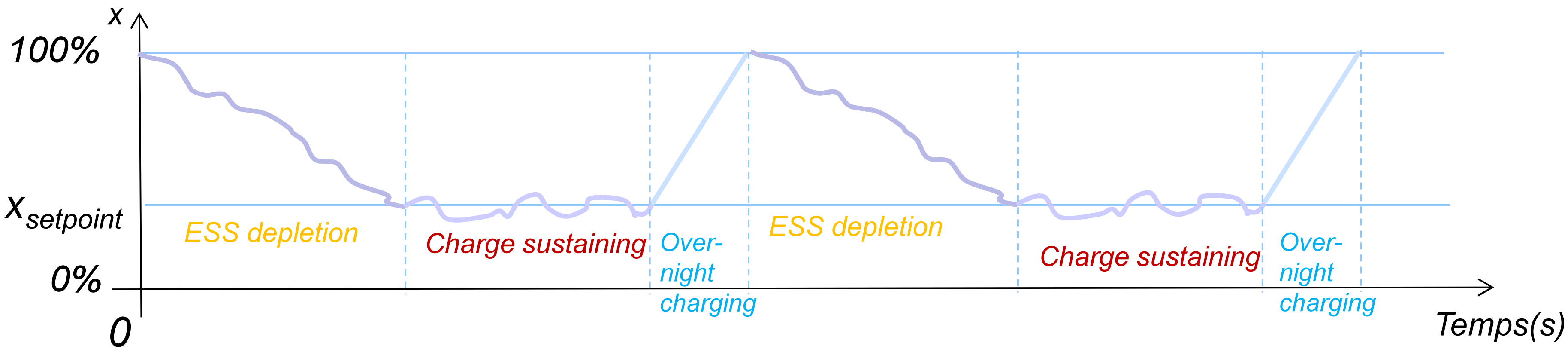
The CISIT Project

The French environment Agency
ADEME

HEV energy management

Energy Storage System (ESS) state management:

- Real time: Charge depletion & sustaining approach (plug-in hybrids)



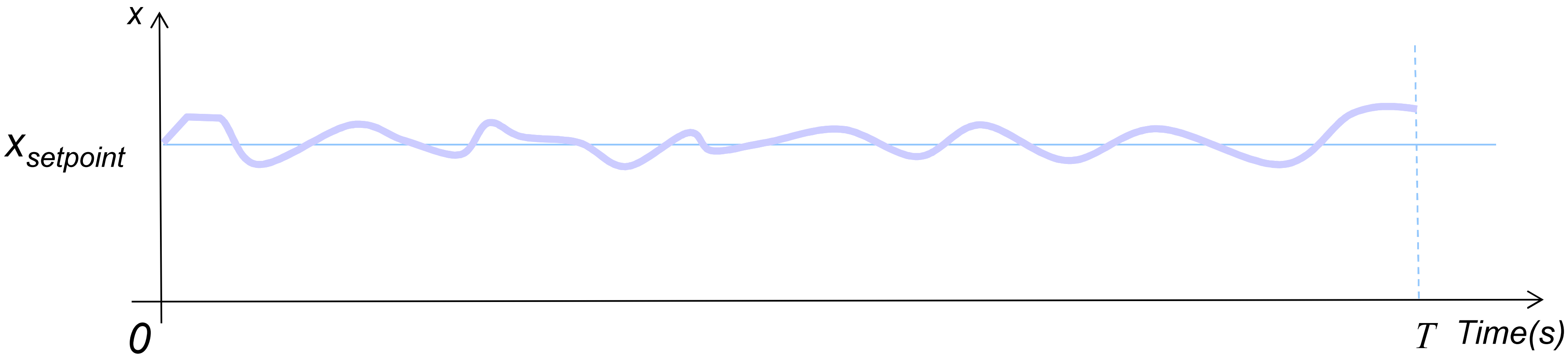
<u>Objective: maximum benefits from a « huge » battery</u>

1) ESS depletion (possibly pure electric mode):
   - No local emission
   - Take care of battery aging
2) Charge sustaining in hybrid mode:
   - Better efficiency than a conventional car
   - Vehicle range similar to conventional car
3) Over-night charging:
   - Low cost
   - Electric network usage

HEV energy management

Energy Storage System (ESS) state management:

- Real time: Charge sustaining approach (micro – full hybrids)



Perfect ESS state control => No battery current

No battery current => No electric machine usage => pure electric mode

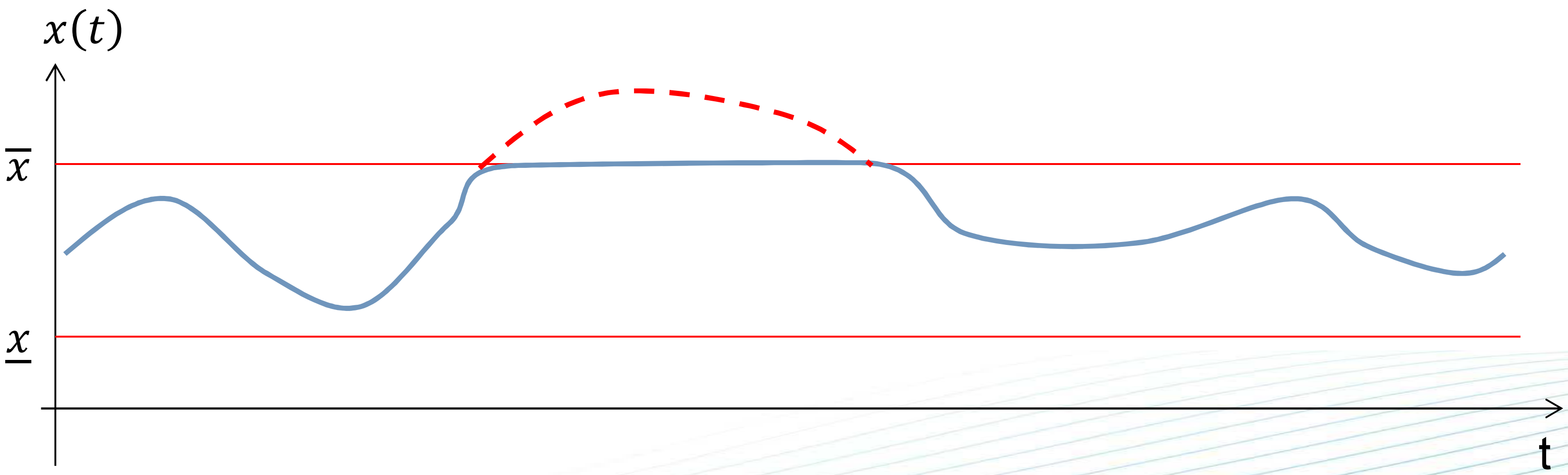
<u>Conclusion :</u> Need a "good" compromise between ESS state regulation and fuel consumption optimization

Introduction

Generally speaking, one should define the desired real time behavior:

- Micro Hybrids
  Small Energy Storage System capacity (e.g. supercapacitor)
  => don't care about the value of the ESS state
  The energy stored is insignificant with respect to the ICE energy

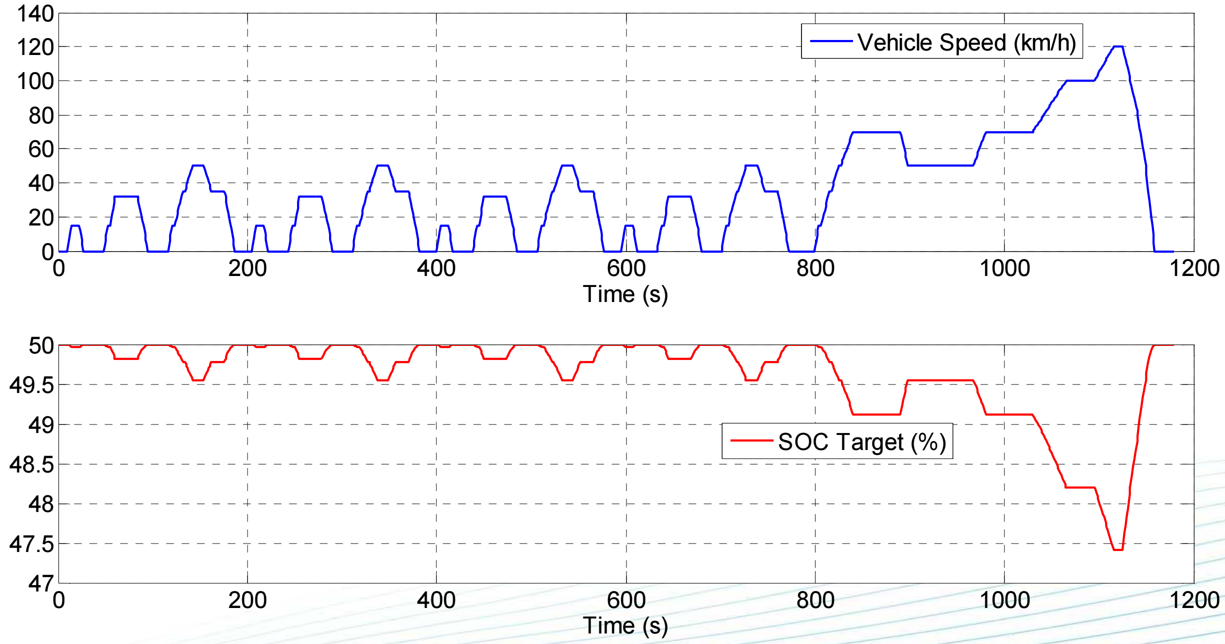
  => BUT ensure that it does not reach the limits



Introduction

Generally speaking, one should define the desired real time behavior:

- Full & Mild Hybrid & (Plug-in in charge sustaining)
  $\Rightarrow$ SOC regulation
  $\Rightarrow$ Careful choice of the SOC target, adapt according to speed



Introduction

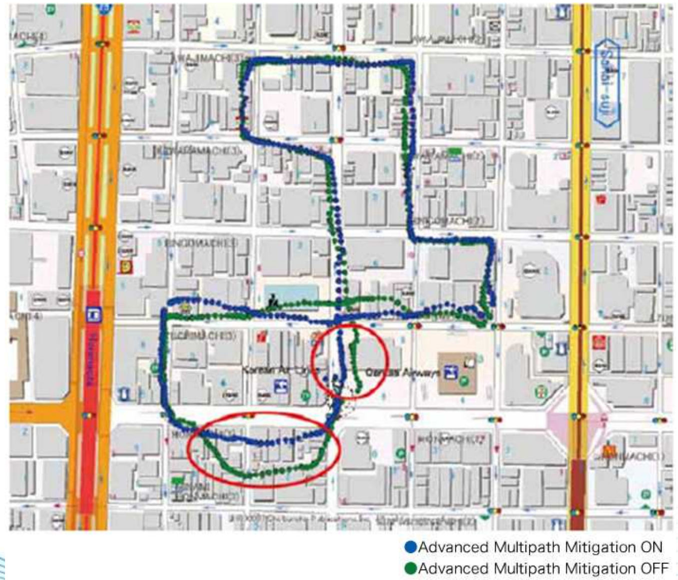
GPS & digital maps

May sounds a good idea, but:

- GPS signal are not really accurate in towns (urban canyon, multi-path)
- Maps helps but may becomes outdated
- The driver should not have to enter his trip before using the car

⇒ The control strategy be able work without GPS

⇒ Nevertheless, even if implementation is difficult, some nice information can be derived



/MACOM's RoadScape/

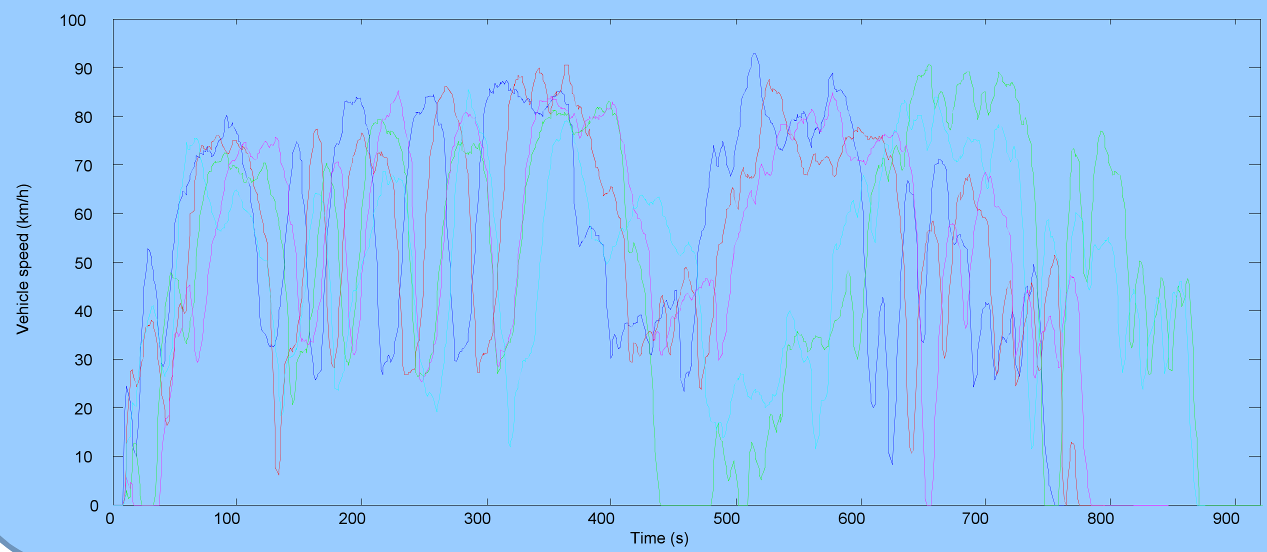
ECMS family

Vehicle Speed Prediction

- It’s difficult to make prediction, especially about the future
- Depends on driver mood, traffic, etc.
- Not only the speed has to be predicted but also the torque

$\Rightarrow$ The acceleration has also to be accurate

Recorded speed, starting from the same position (need GPS)



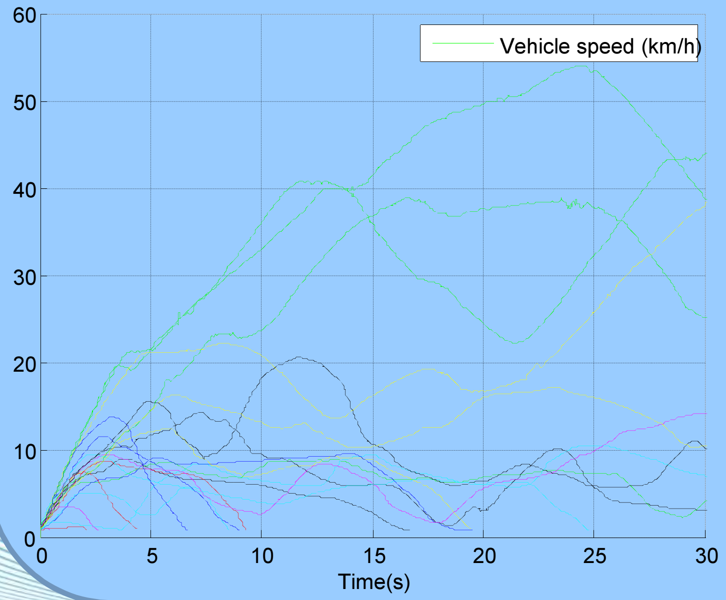
ECMS family

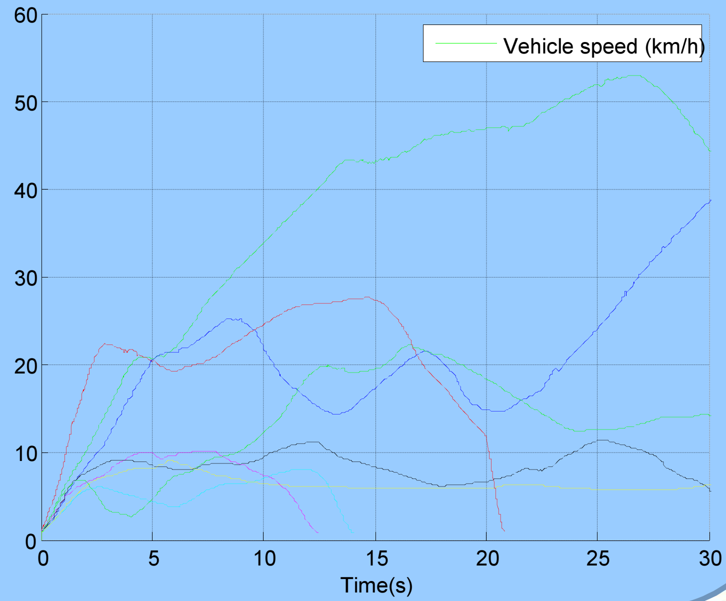
Vehicle Speed Prediction

- It's difficult to make prediction, especially about the future
- Depends on driver mood, traffic, etc.
- Not only the speed has to be predicted but also the torque

$\Rightarrow$ The acceleration has also to be accurate

Recorded speed, starting from the random position (no GPS)





Introduction

An optimal control algorithm is available in simulation

Question :
Is it possible to use it, somehow, to derive (possibly efficient) suboptimal real time controls ?

2 main approaches:

- Given some simulation results, one could try to extract some "knowledge" from this results to "mimic" the optimal control behavior => *Regression based algorithms*

- Given the PMP optimal control algorithm, adapt it so it can be used in real time => *ECMS family*

Introduction

An optimal control algorithm is available in simulation

Question :
Is it possible to use it, somehow, to derive (possibly efficient) suboptimal real time controls ?

2 main approaches:

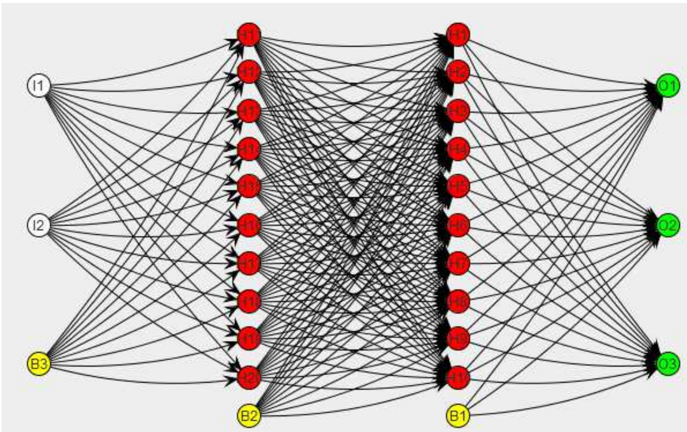
- Given some simulation results, one could try to extract some "knowledge" from this results to "mimic" the optimal control behavior => *Regression based algorithms*
- Given the PMP optimal control algorithm, adapt it so it can be used in real time => *ECMS family*

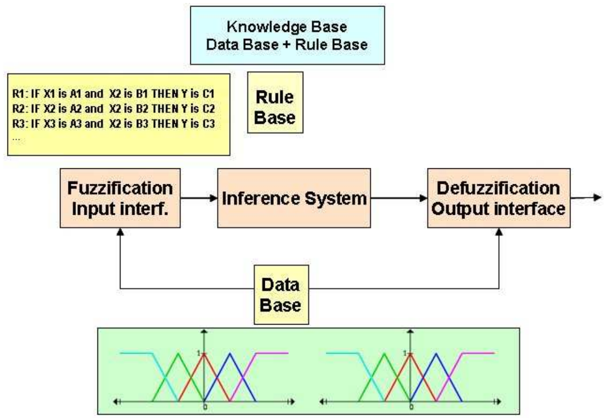
Introduction

1st approach : Regression over optimal data set

<u>Idea:</u> The optimal control results explain how the real time controller should react to some driving conditions.
If optimal results are generated for *all possible* driving conditions, then these results contains the whole knowledge about the HEV optimal control.

<u>How to capture this "knowledge":</u> Neural Network, Fuzzy systems





/ www.heatonresearch.com/

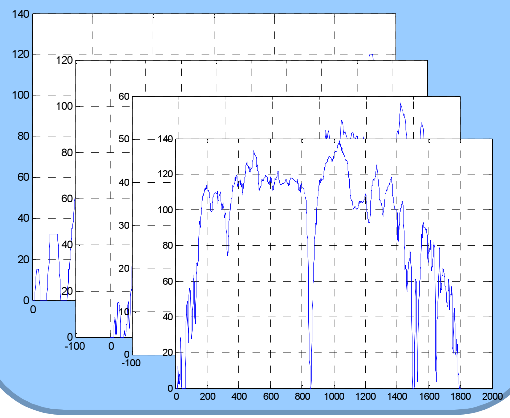
Regression based algorithms

VPPC 2014

<u>Step 1 :</u> generate optimal control for a wide range of driving cycle & Initial Soc

/Scordial & al. 2009/ / Taghavipour, Foumani, & Boroushaki, 2012/

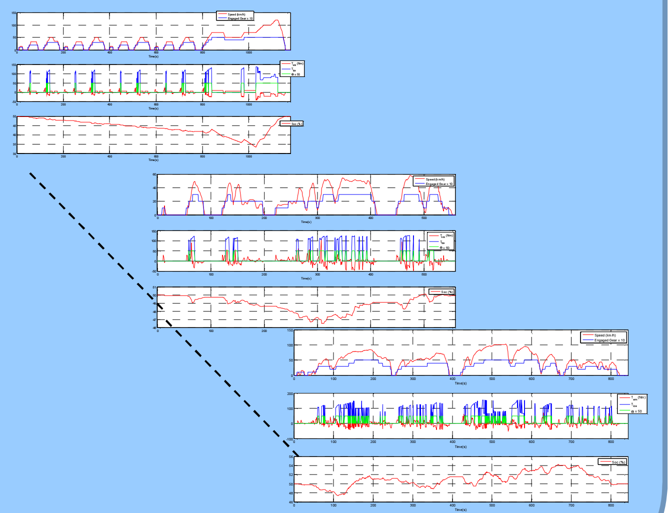
Set of n_{cycles} driving cycles



Set of n_{soc} initial Soc:
$0, 0.1, 0.2, \ldots, 0.9, 1$

Optimal control
(Simulation)

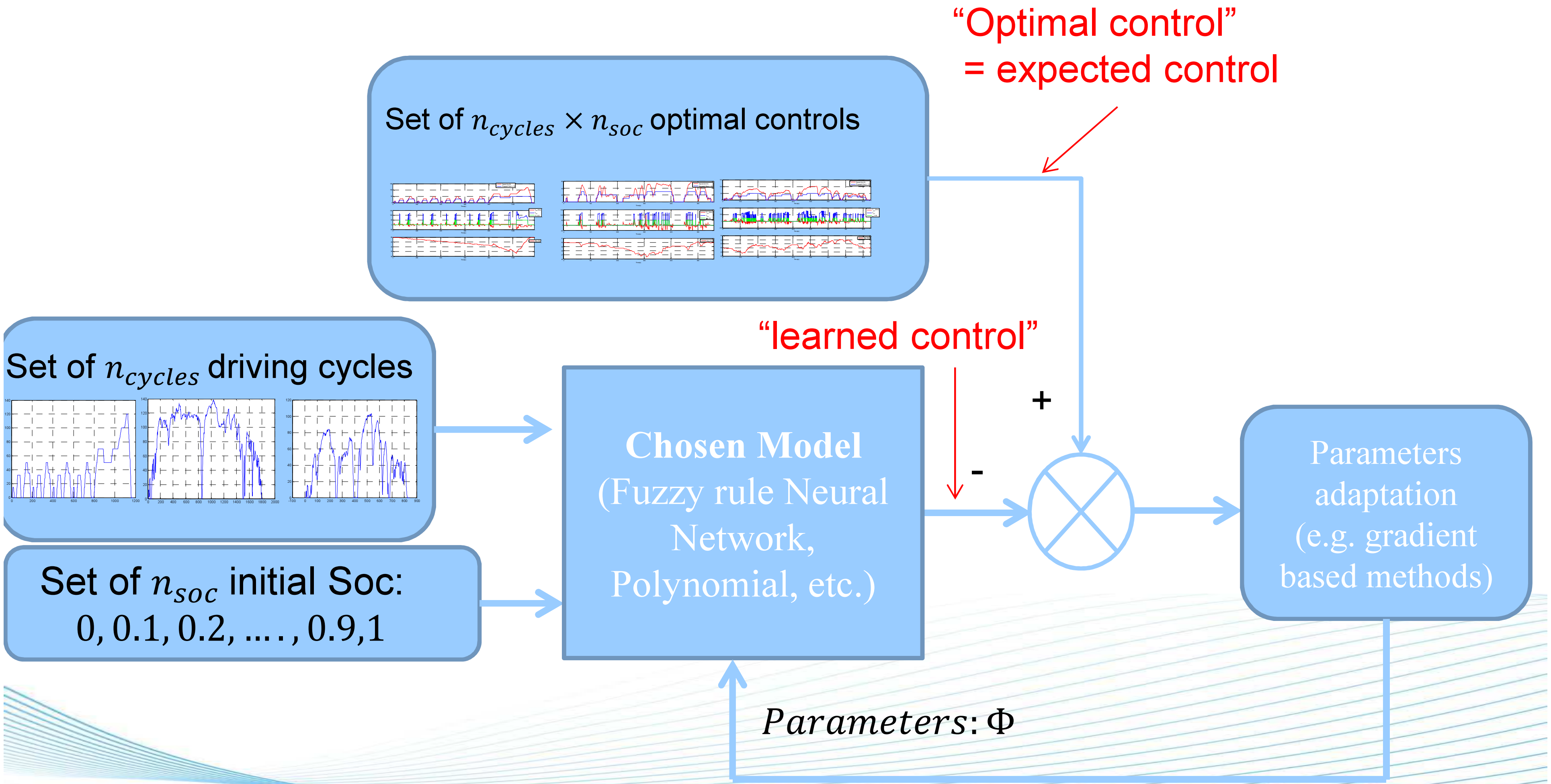
Set of $n_{cycles} \times n_{soc}$ optimal controls



Regression based algorithms

<u>Step 2 :</u> Chose an arbitrary control structure and perform off-line regression

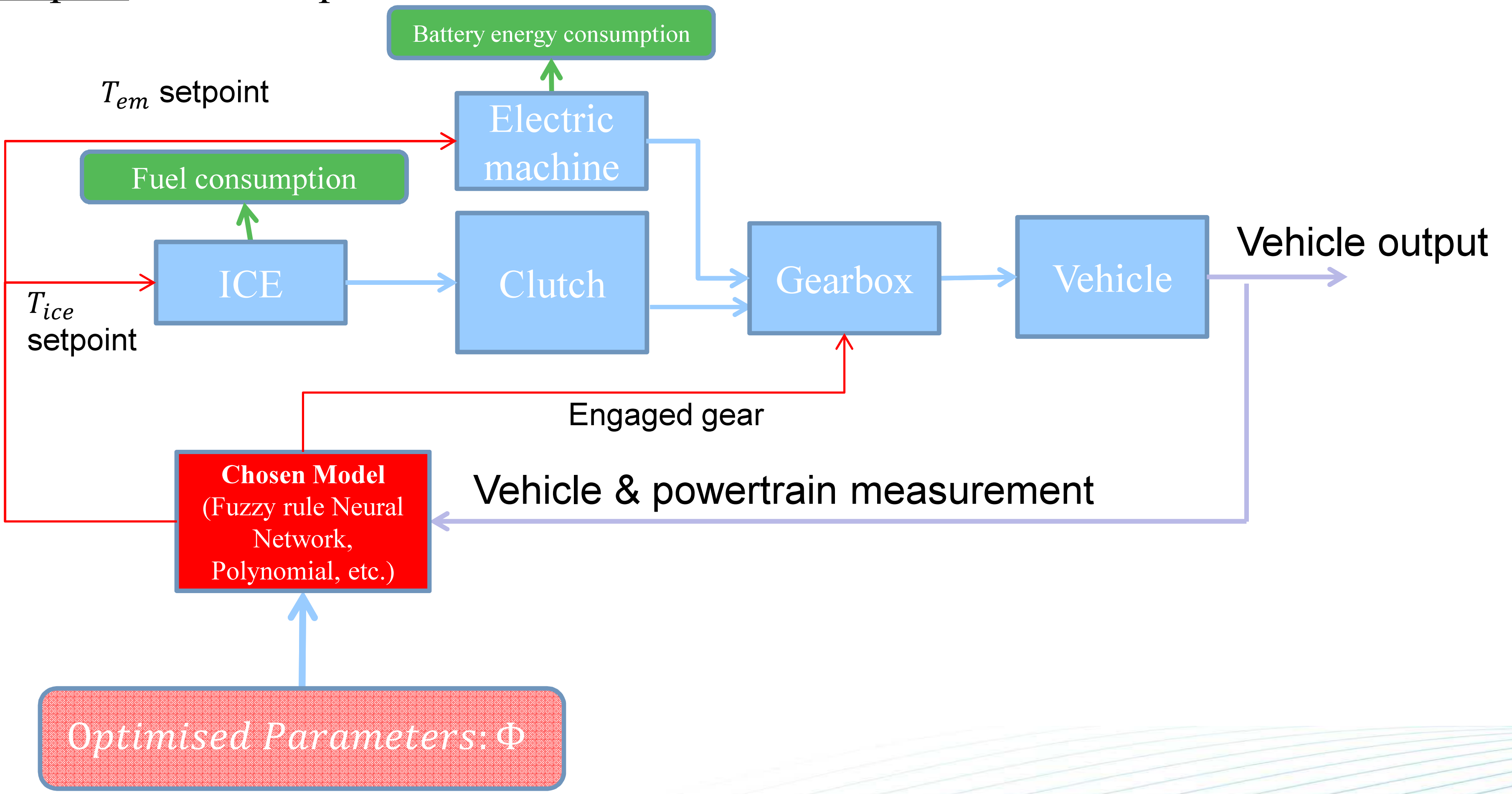
/Scordial & al. 2009/ / Taghavipour, Foumani, & Boroushaki, 2012/



Regression based algorithms

/Scordial & al. 2009/ / Taghavipour, Foumani, & Boroushaki, 2012/

Step 3 : Online implementation



Regression based algorithms

<u>Regression based controls:</u>

- Systematic approach
- Can work with many arrangements provided an optimal control algorithm is available

BUT

- No guaranty on the real time performance
- The optimal data set may not fulfill some theoretical requirements
  => Boolean variable switches captured with a polynomial
- Need to consider many driving situations to cover all the possible driving situations

Summary

Part 2 : Real time control

I) Introduction

II) Regression based algorithms

III) ECMS Family

ECMS family

VPPC 2014

- Let us recall: $u^*(t) = \underset{v \in \Phi(W(t))}{\arg\min} H\left(t, v, x^*(t), W(t), \lambda^*(t)\right)$

$$H(t,u,x,W,\lambda) = \underbrace{Q(t,u,W,x_0)}_{\text{Fuel consumption}} + \overbrace{\lambda^T}^{\text{Co-state}} \cdot \underbrace{f(t,u,W,x)}_{\text{Electrical consumption}}$$

Minimizing the Hamiltonian is simple and can be done online:

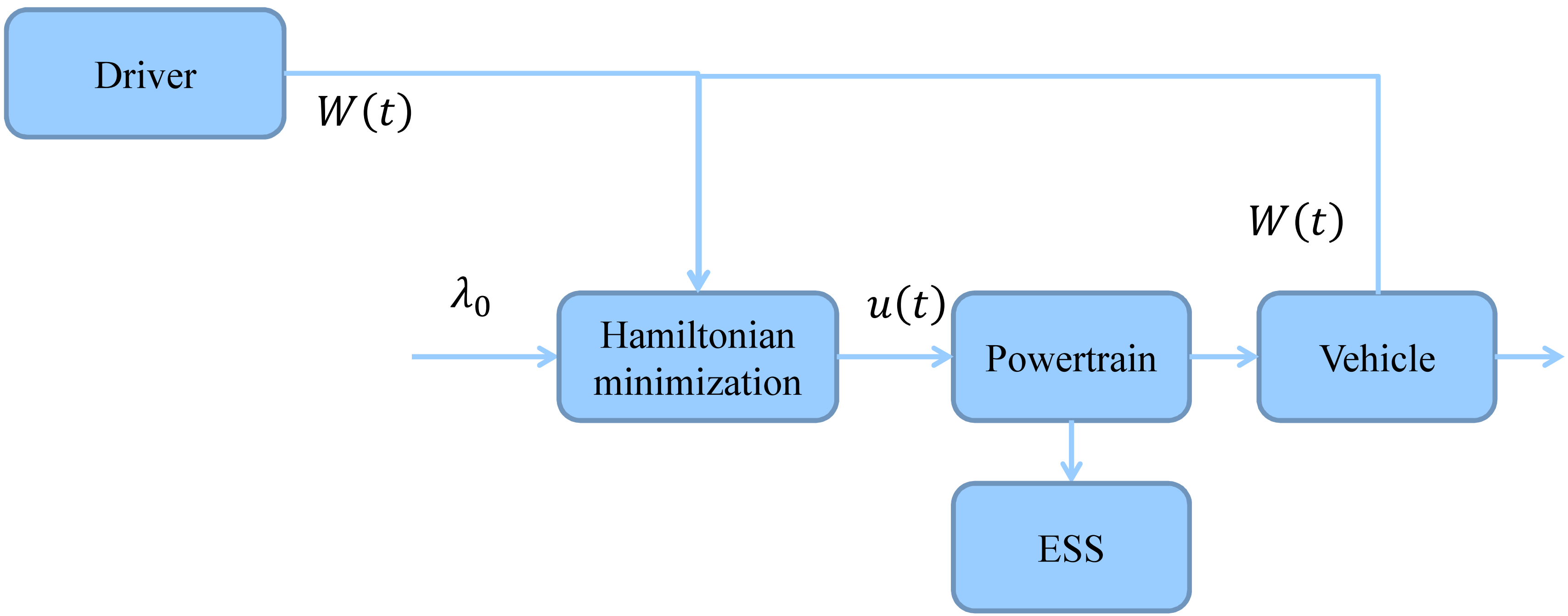
=> instantaneous optimization of the electrical & fuel consumption

<u>So what's the problem ???</u>

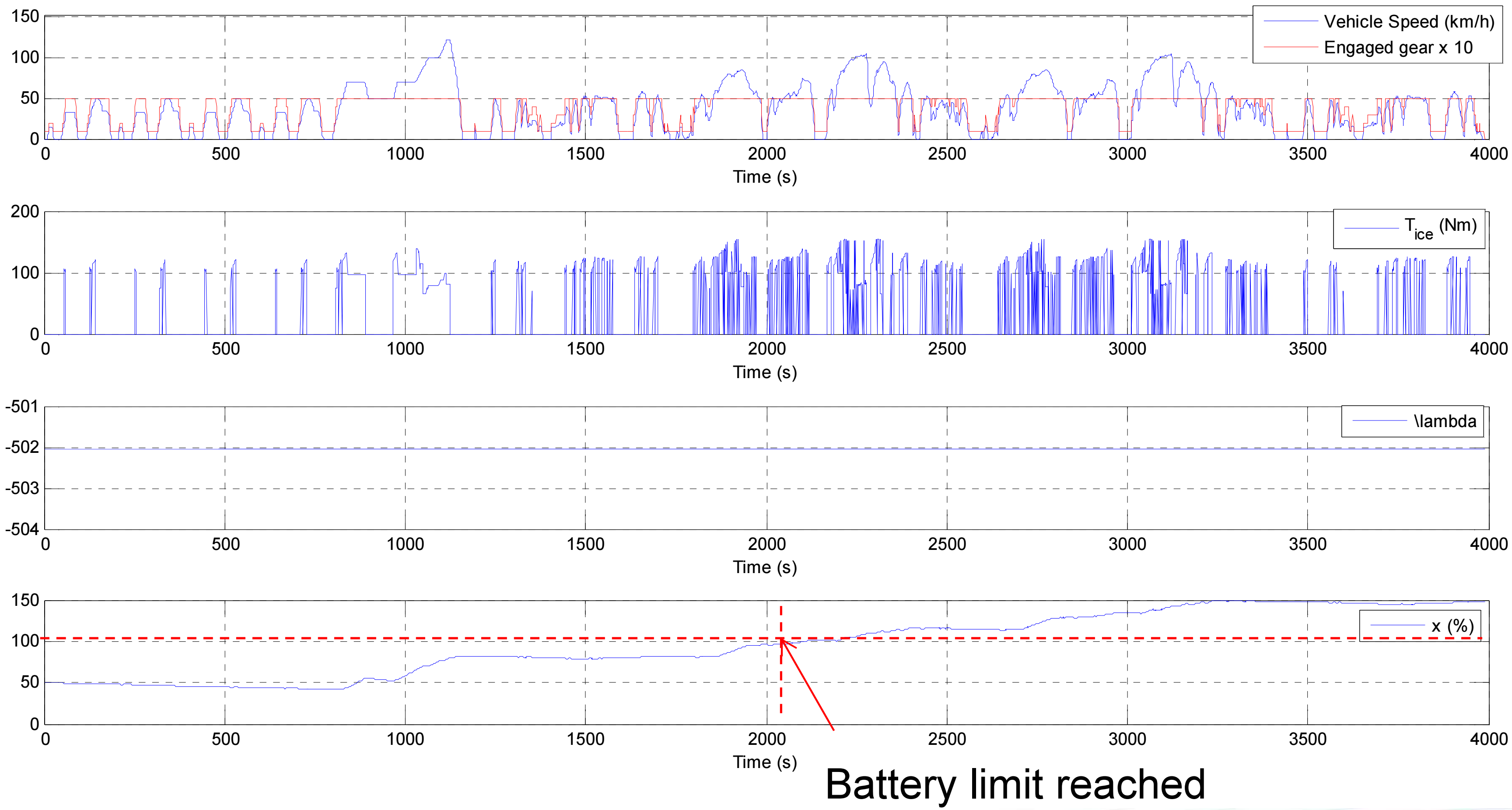
- Initial co-state value is unknown
- Future driving conditions are unknown => co-state cannot be computed
- State trajectory will diverge and reach the physical and/or safety limits
- The co-state need to be controlled

ECMS family

- Basic control scheme



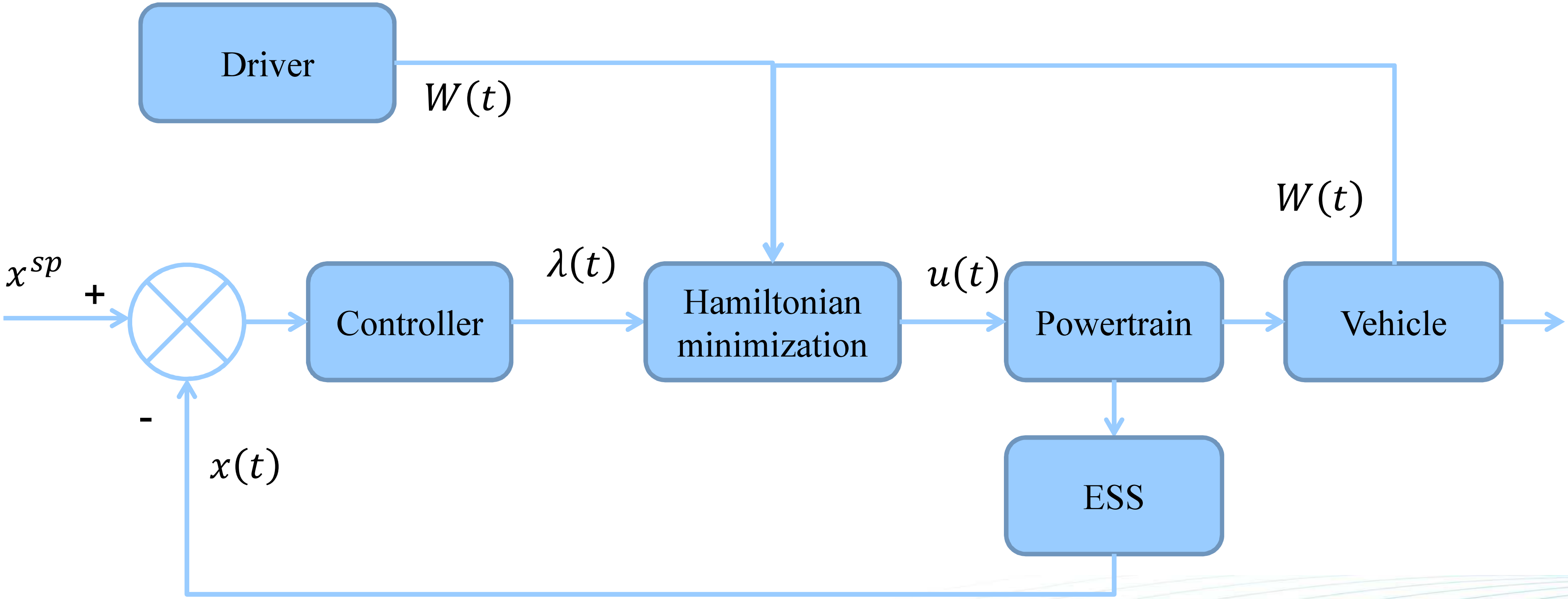
ECMS family



Obviously keeping the co-state constant is quite dangerous.
Simplest solution : ESS state control

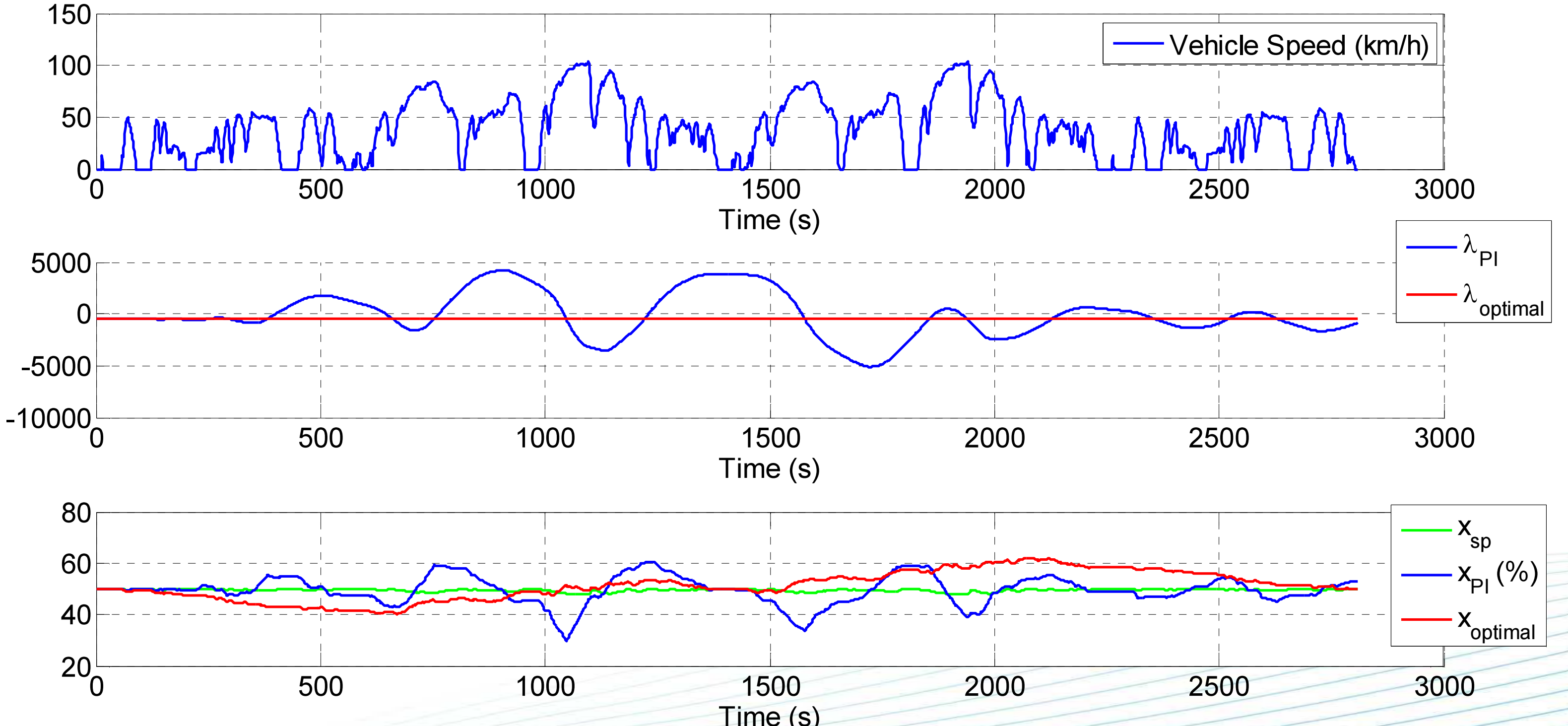
ECMS family

- Basic control scheme : ESS state control



Real time control

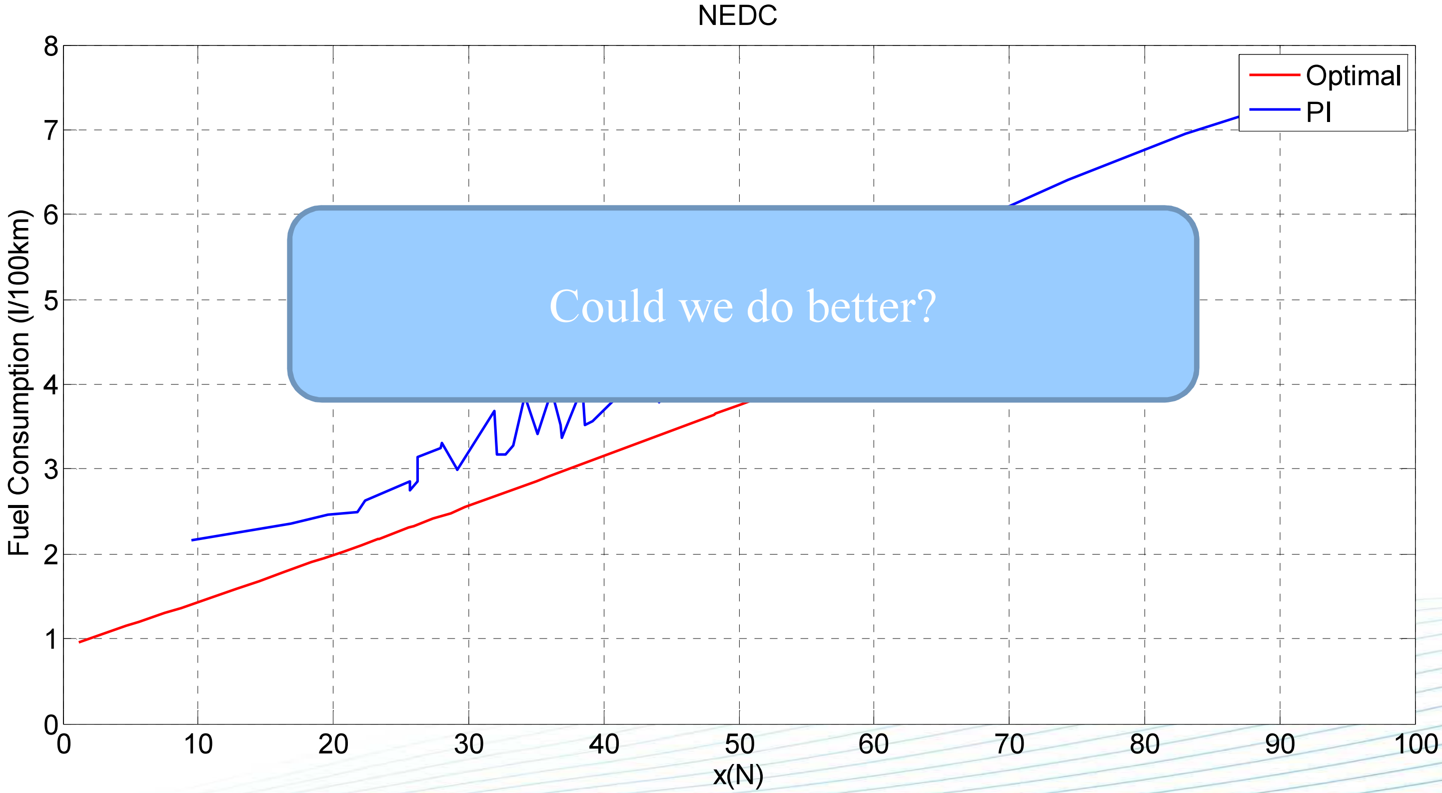
Arbitraty initial value : $\lambda(0) = -500$
Tuning : slow dynamics
PI : Fuel consumption : 5,04 l/100 km $x(N) - x(0) = 3{,}2\%$
Optimal: $\lambda_0 = -478$ Fuel consumption : 4,49 l/100 km $x(N) - x(0) = 0{,}12\%$



ECMS family

Control strategy performances analysis:

- Need to account for both energy sources : conso=f(soc)
- Mean distance between both curves (24%)



Could we do better?

Predictive control

Discrete time implementation:

- Sampled data (piecewise constant)
- Fixed step Euler solver

Continuous time

- System Dynamics :

$$\dot{x}(t) = f\big(t, u(t), W(t)\big)$$

- Co-state dynamics:

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x}\big(t, u(t), W(t), \lambda(t)\big)$$

$$\dot{\lambda}(t) = 0$$

- Hamiltonian minimization:

$$u^*(t) = \arg\min_{v \in \Phi(W(t))} H\big(t, v, x(t), W(t), \lambda(t)\big)$$

Euler →

Discrete time

- System Dynamics :

$$x(i+1) = x(i) + f\big(\mathrm{i} \cdot \mathrm{s}, u(i), W(i)\big)$$

- Co-state dynamics:

$$\lambda(i+1) = \lambda(i) - \frac{\partial H}{\partial x}\big(t, u(i), W(i), \lambda(i)\big) \cdot s$$

$$\lambda(i+1) = \lambda(i) = \lambda_0$$

- Hamiltonian minimization:

$$u^*(i) = \arg\min_{v \in \Phi(W(\mathrm{i}))} H\big(\mathrm{i} \cdot s, v, x(\mathrm{i}), W(\mathrm{i}), \lambda(\mathrm{i})\big)$$

Predictive control

<u>State trajectory:</u>

- System Dynamics :

$$x(t) = x_0 + \int_0^T f\left(t, \Pi\left(W(t), \lambda(t)\right), W(t), x(t)\right) \cdot dt$$

$$x(i) = x_0 + \sum_{i=0}^{i-1} f\left(t, \Pi\left(W(i), \lambda(i)\right), W(i), x(i)\right) \cdot s$$

=> So everything is more or less the same as for the continuous case

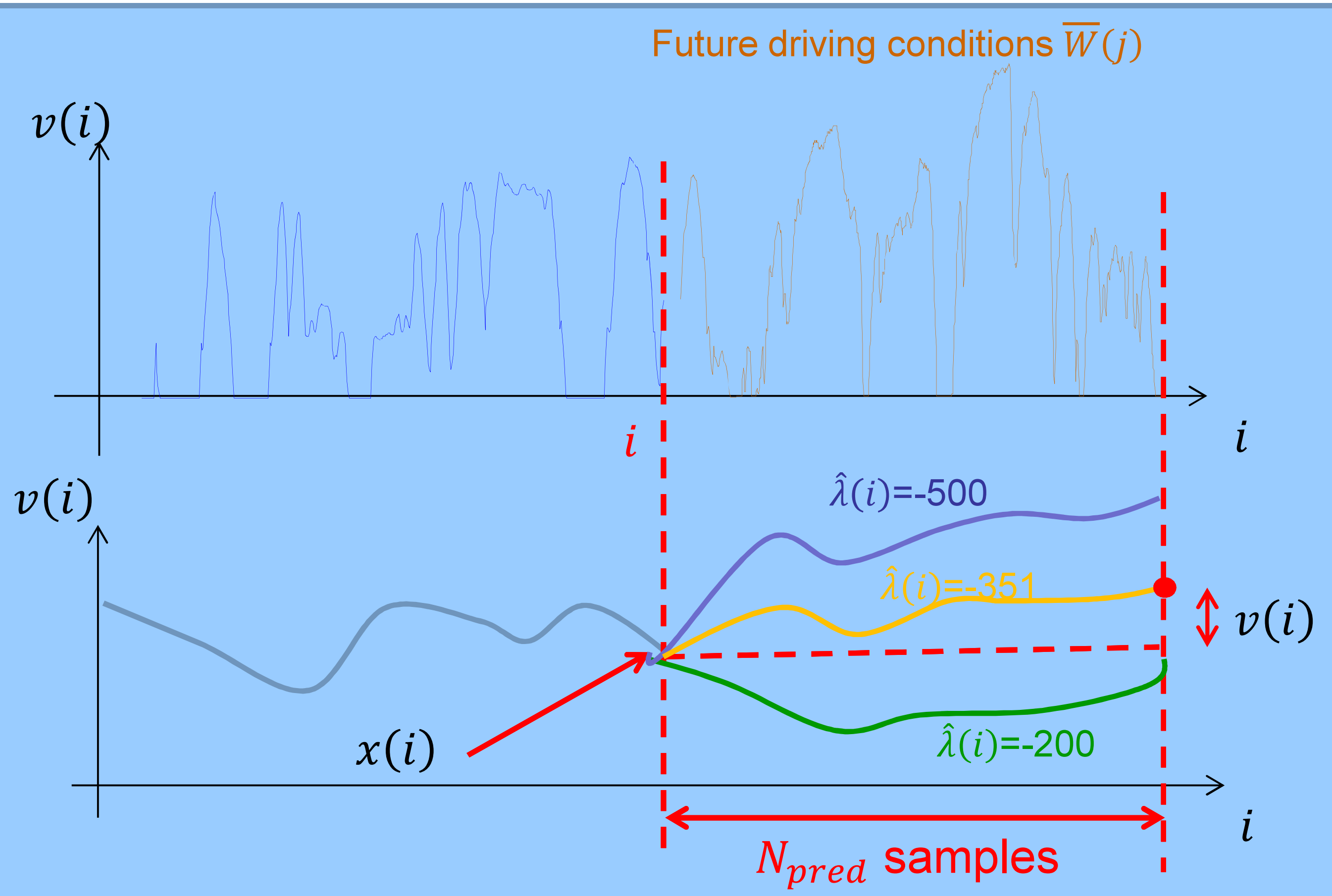
<u>Let us recall:</u>

- If the future driving are known, then the co-state can be computed,
- then we could have an optimal real time algorithm

Predictive control

Embed the optimization algorithm within a predictive framework:

- Future values are denoted with a bar
- Initial state value at the beginning of the horizon : $x(i)$ (known value)
- Fix an arbitrary expected final state value: $\bar{x}(i+N_{pred}) = x(i) + v(i)$



Predictive control

Embed the optimization algorithm within a predictive framework:

- Future values are denoted with a bar
- Initial state value at the beginning of the horizon : $x(i)$ (known value)
- Fix an arbitrary expected final state value: $\bar{x}(i+N_{pred}) = x(i) + v(i)$

<u>At every sampling time *i*</u>

Horizon : $J=[i, i+N_{pred}]$

System : $\bar{x}(j+1) = \bar{x}(j) + f\left(t, \Pi\left(W(j), \hat{\lambda}(i)\right), \bar{W}(j)\right) \cdot s$

Criterion : $J = \min \sum_{j \in J} Q\left(\bar{u}(j), \bar{W}(j)\right) s$

Constraints :

$\bar{u}(j) \in \mathrm{U}\left(\bar{W}(j)\right)$

$\bar{x}(i) = x(i)$

$\bar{x}(i+N_{pred}) = x(i) + v(i)$ ← Desired SOC increase at sample time *i*

Predictive control

Let $\hat{\lambda}(i)$ be the (initial) co-state solution to the i[th] optimization problem

At each sampling time, the control is only a function of:

- The co-state
- The exogenous variables :

$$u(j) = \Pi\left(\overline{W}(i), \hat{\lambda}(i)\right) \qquad j \in J = \left[i, i + N_{pred}\right]$$

According to the systems dynamics, the corresponding SOC variation is :

$$M(W, \lambda) = f\left(t, \Pi(W, \lambda), W\right)$$

Over the optimisation Horizon $J = [i, i + N_{pred}]$, the state of charge depends only on $\hat{\lambda}(i)$ and $\overline{W}(j)$:

$$\bar{x}\left(i + N_{pred}\right) = x(i) + \sum_{j \in J} M\left(\hat{\lambda}(i), \overline{W}(j)\right) \cdot s$$

Predictive control

Finally, if the future driving conditions are known, the problem can be solved

$$g\left(\hat{\lambda}(i)\right) = \sum_{j \in J} M\left(\hat{\lambda}(i), \bar{W}(j)\right) \cdot s$$

$$\bar{x}\left(i + N_{pred}\right) = x(i) + v(i) \Leftrightarrow \underbrace{g\left(\hat{\lambda}(i)\right) = v(i)}_{\text{Shooting algorithm}}$$

How to deal with prediction :

- Implement complex prediction scheme
  => accuracy ok only during 20 seconds /Musardo & al. 2005/
- Use some information about planned journey
- Predict something "*easier*" to predict

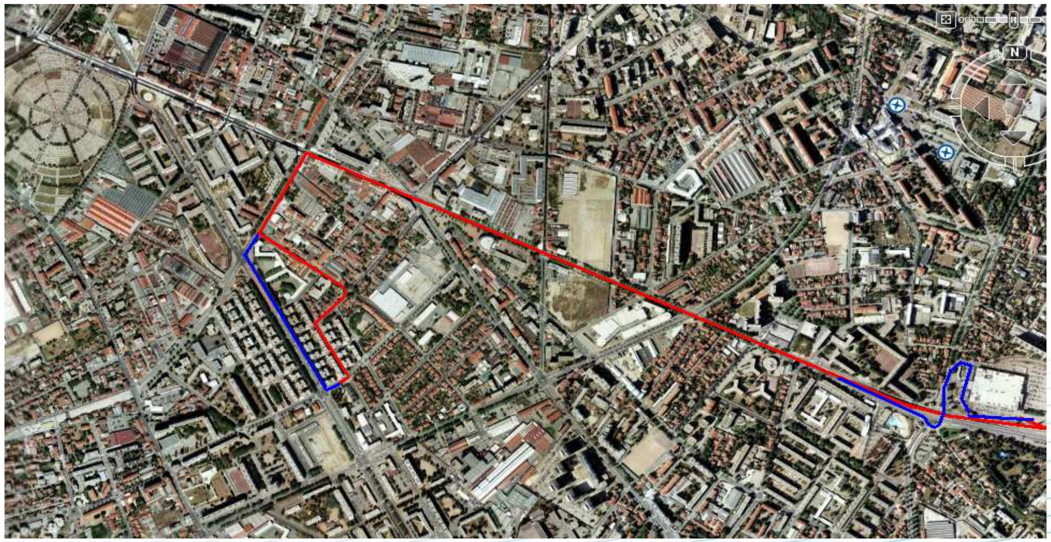
Vehicle on predefined routes

Vehicles on a predefined route: Bus, delivery trucks, cleaning machines, etc.

- The vehicle itinerary is known
- Traffic is a disturbance
- How can we use this information ?

IFSTTAR

/S. Kermani PhD Thesis/

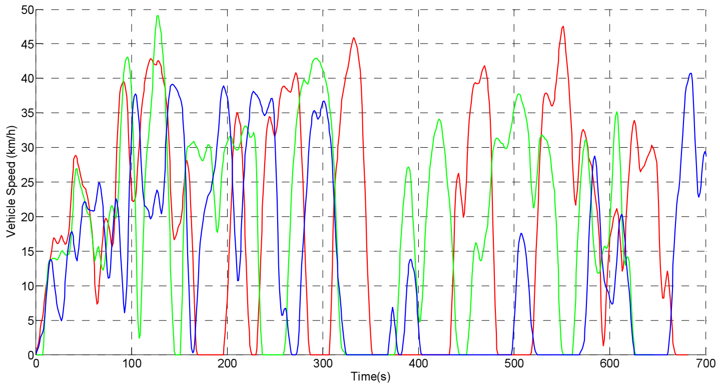


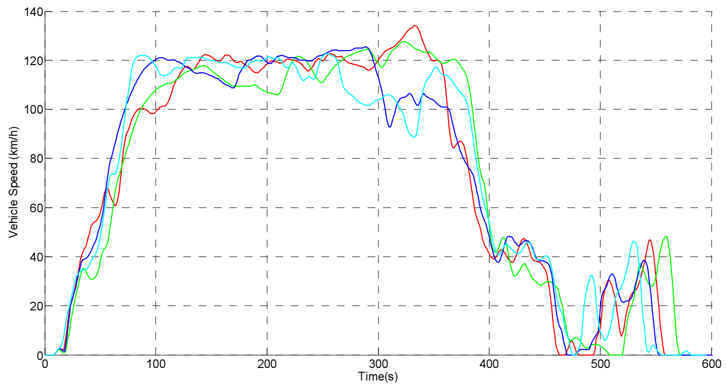
Urban trip (Bron, France)

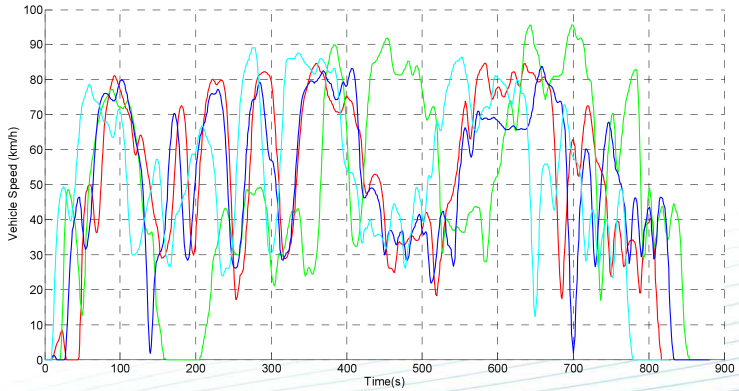
Vehicle on predefined routes

Vehicles on a predefined route: Bus, etc.

- The vehicle itinerary is known
- Traffic is a disturbance
- How can we use this information ?



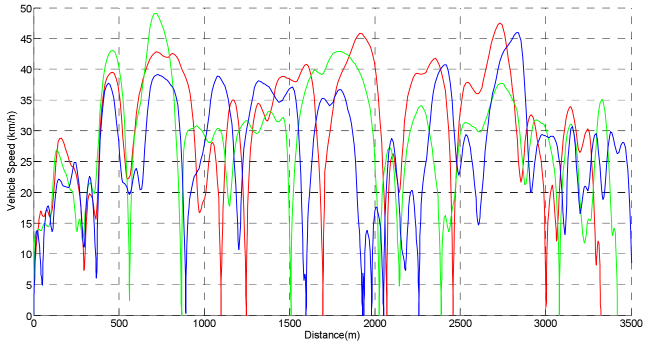


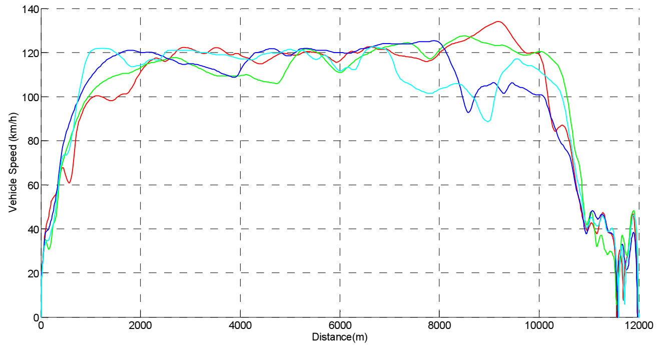


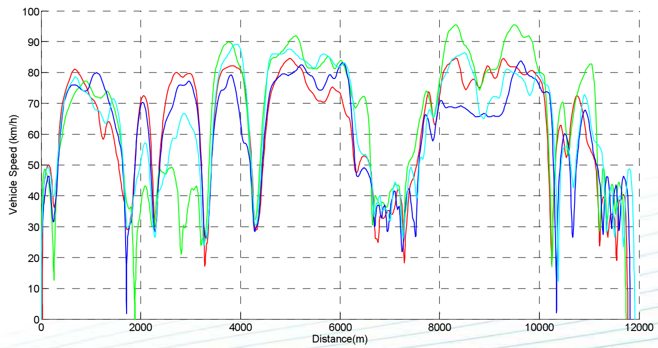
Vehicle on predefined routes

Vehicles on a predefined route: Bus, etc.

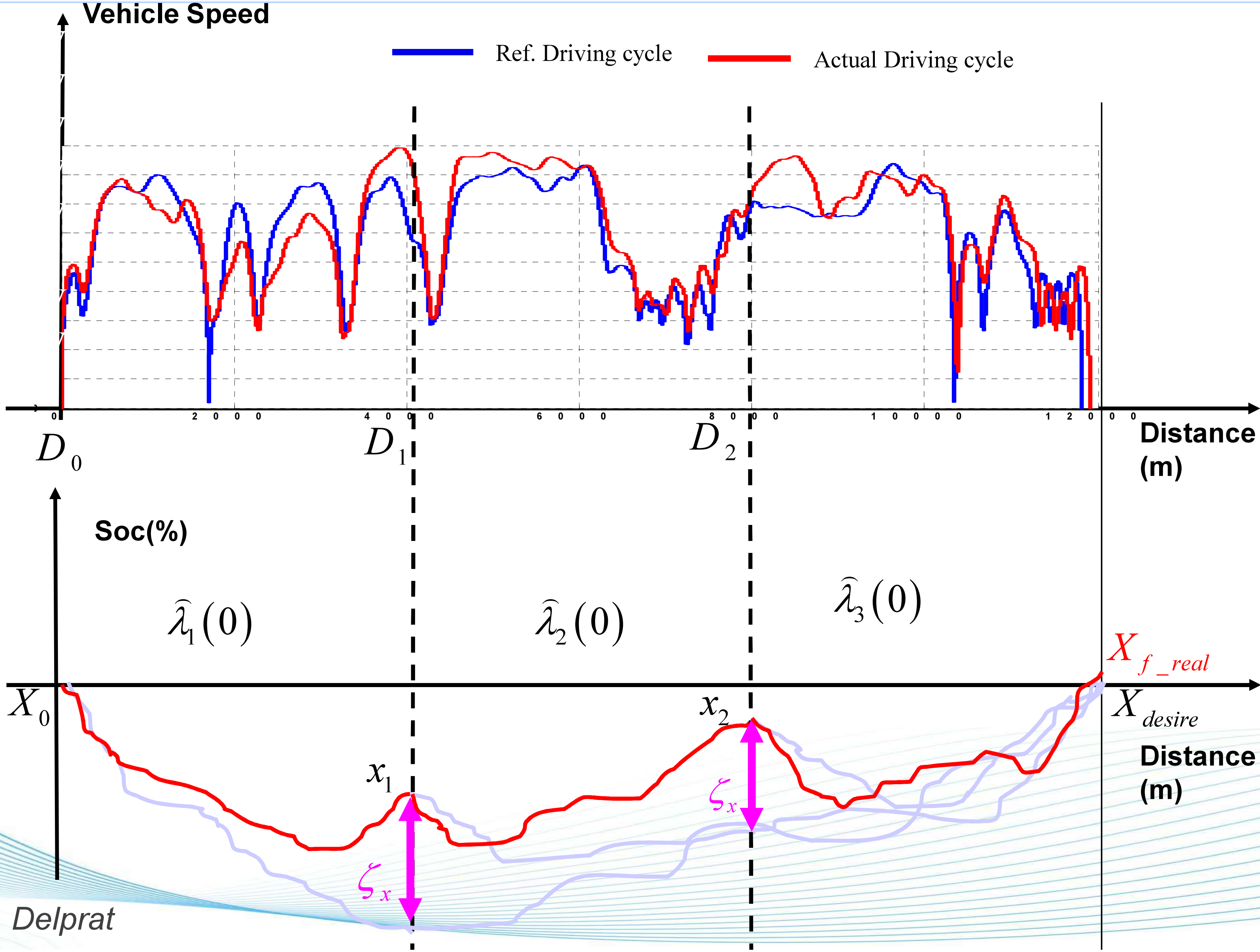
- The driver react to environment & traffic
- Environment (stop sign, traffic lights, speed limits, etc.) is fixed
- Consider distance as the reference variable







Vehicle on predefined routes



S. Delprat

Vehicle on predefined routes

The considered mild hybrid vehicle:

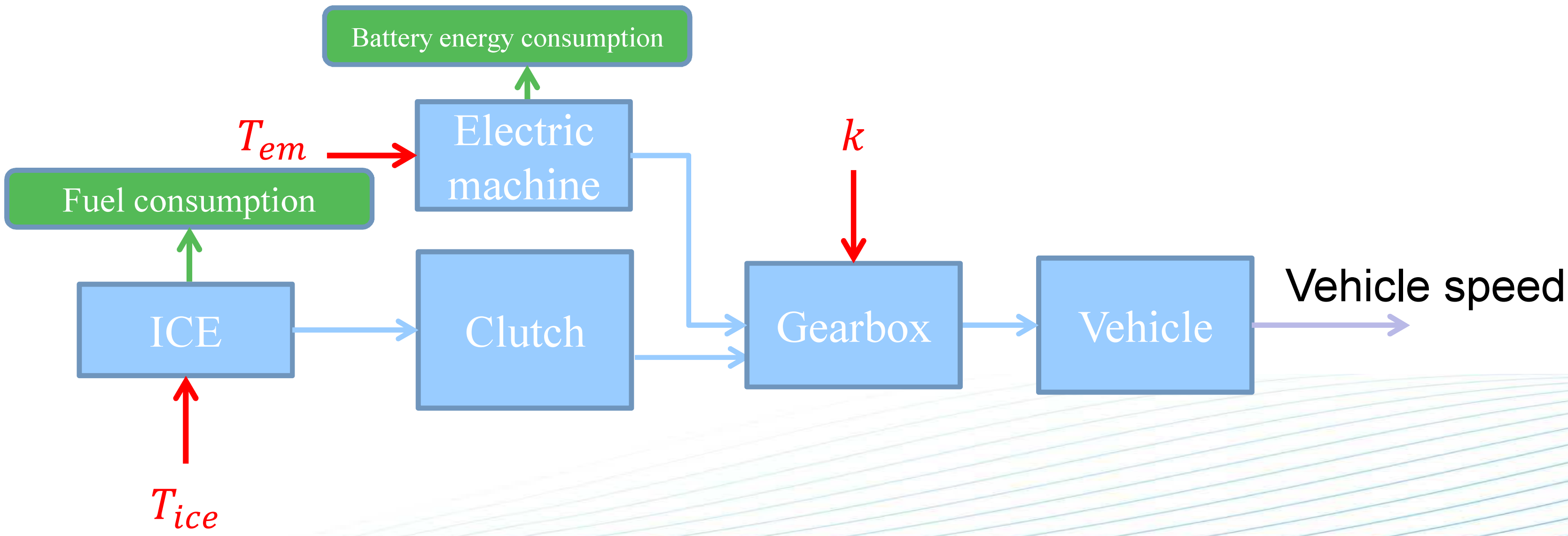
- Mass : 800 kg


=60kW
Diesel

+15kW

+42V/34Ah NiMh

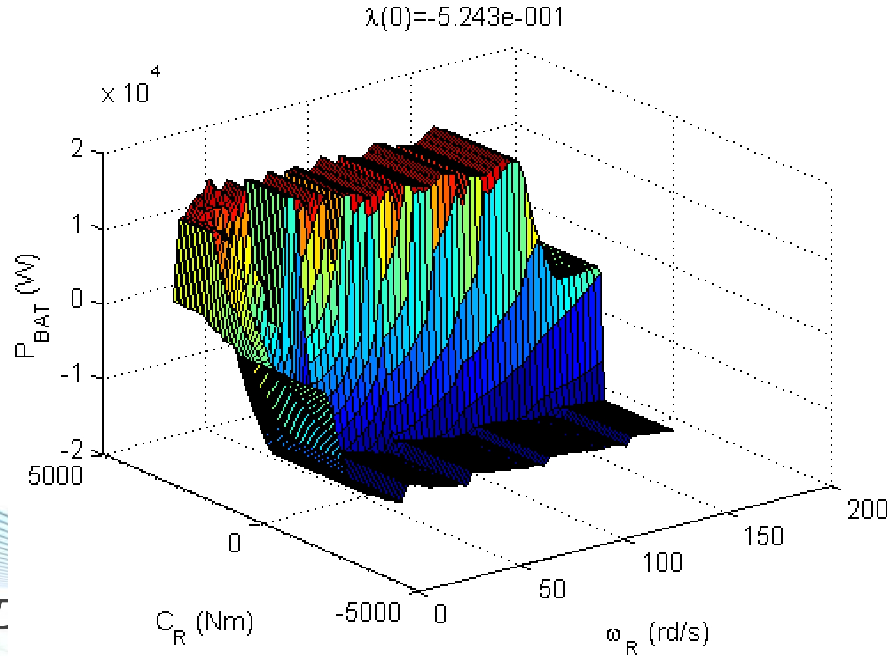
+ 5 Gears Gear box

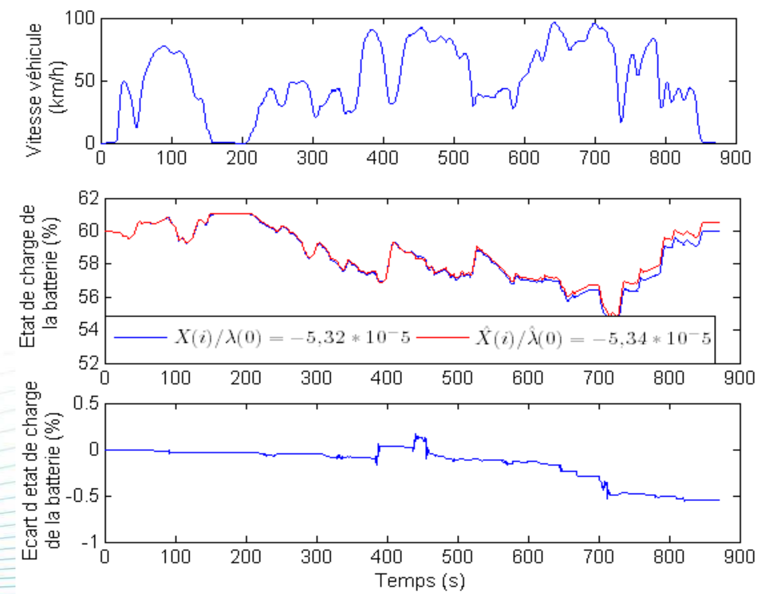


Vehicle on predefined routes

Implementation details:

- The future driving conditions are given by a reference driving cycle
- The prediction horizon starts, on the reference driving, at the actual covered distance
- Pre-compute the battery usage on a 4D maps $\bar{x}\left(i+N_{pred}\right)=x(i)+\sum_{j\in J} M\left(\hat{\lambda}(i),\bar{W}(j)\right)\cdot s$



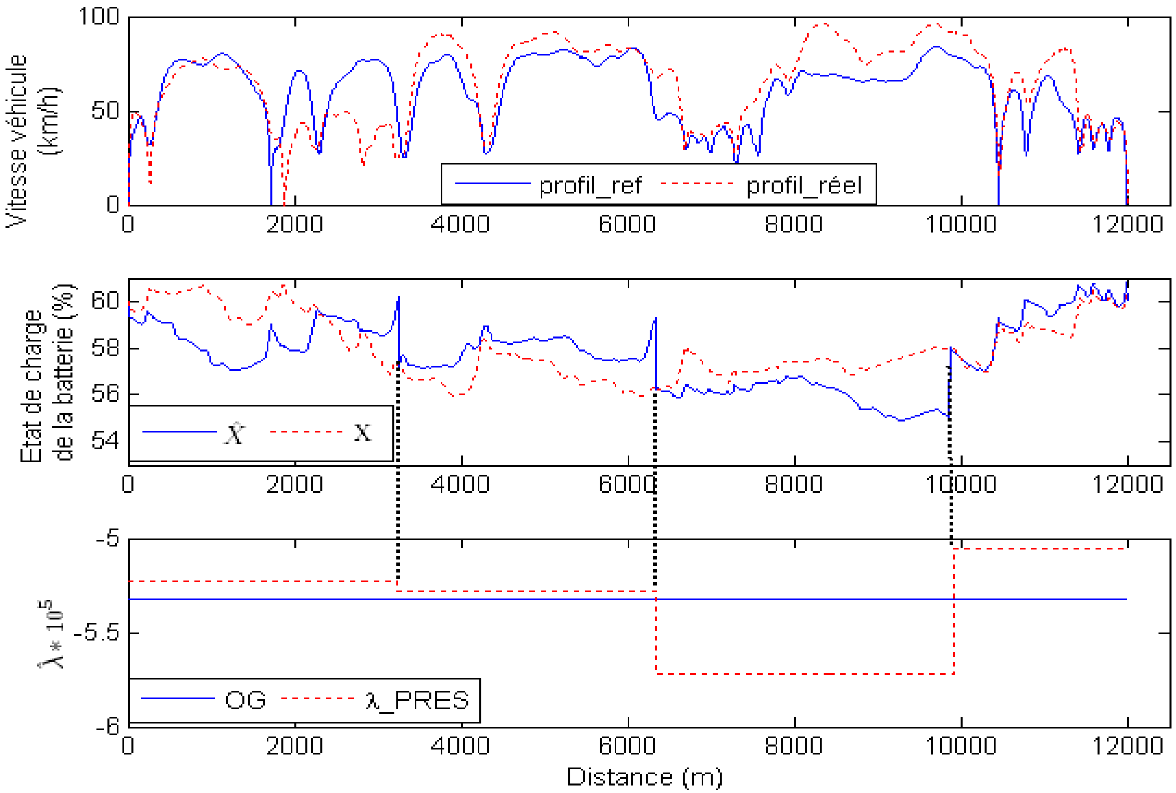


Vehicle on predefined routes

λ_{PRES} control strategy:
tuning Paramters:

- Sampling Perdiod s
- SOC deviation threshold : $\zeta_x = 3\%$

$x(N) - x(0) = 0{,}03\%$ *Fuel cons. = 2,94 l/100km vs Optimal : 2,93 l/100km Deviation : 0,5%*



Vehicle on predefined routes

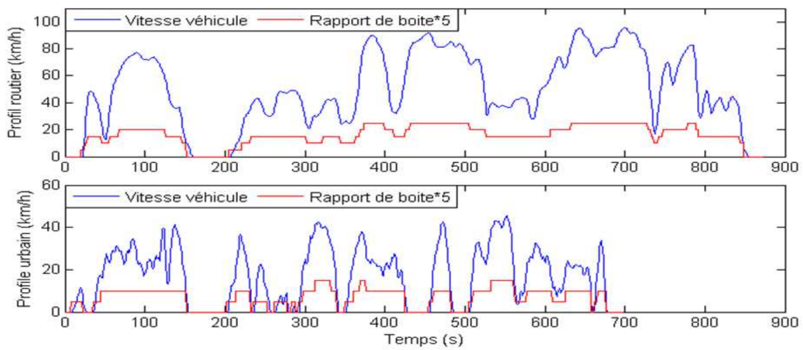
Experimental results at the IFSTTAR

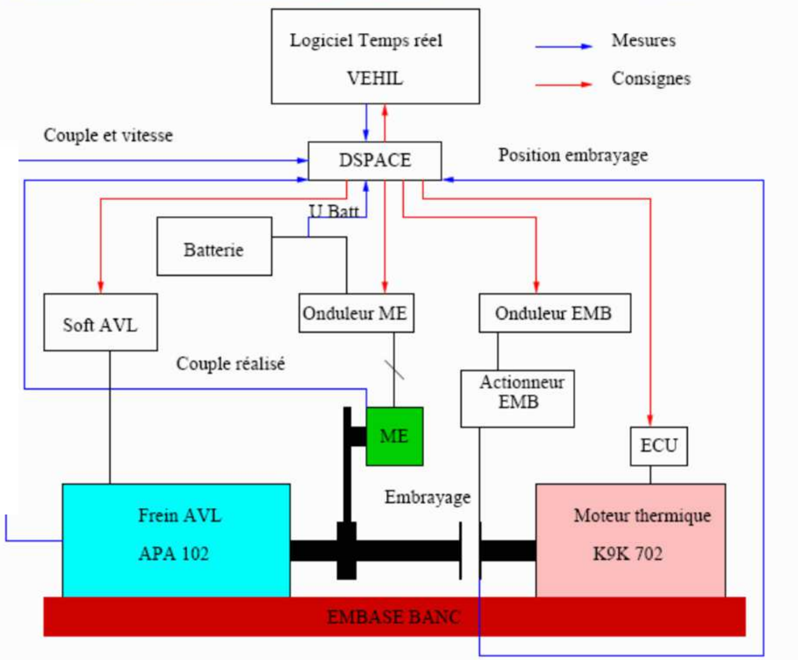
/S. Kermani PhD Thesis/

Actual 1,5L Diesel engine 60 kW
Emulated vehicle : Clio II
Euro III
2 routes, SOC target variations

Road

Urban





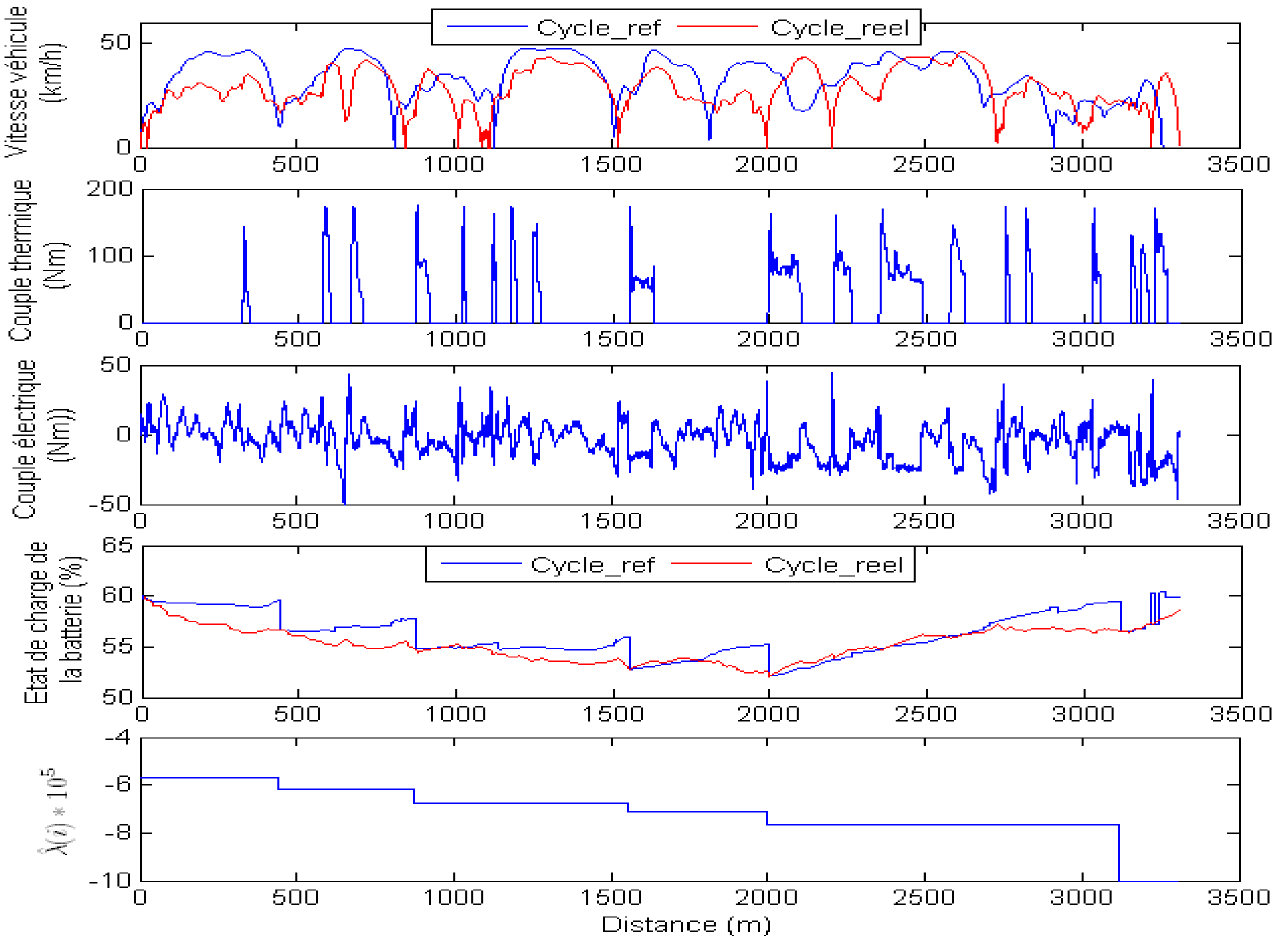
Vehicle on predefined routes

VPPC 2014

Experimental results:

- The algorithm can be executed in real time
- One of the experimental driving cycle is chose as reference
- A second driving cycle is used for the experiment

$x(N) - x(0) = -1{,}52\ \%$ *Fuel cons. = 3,16 l/100km*

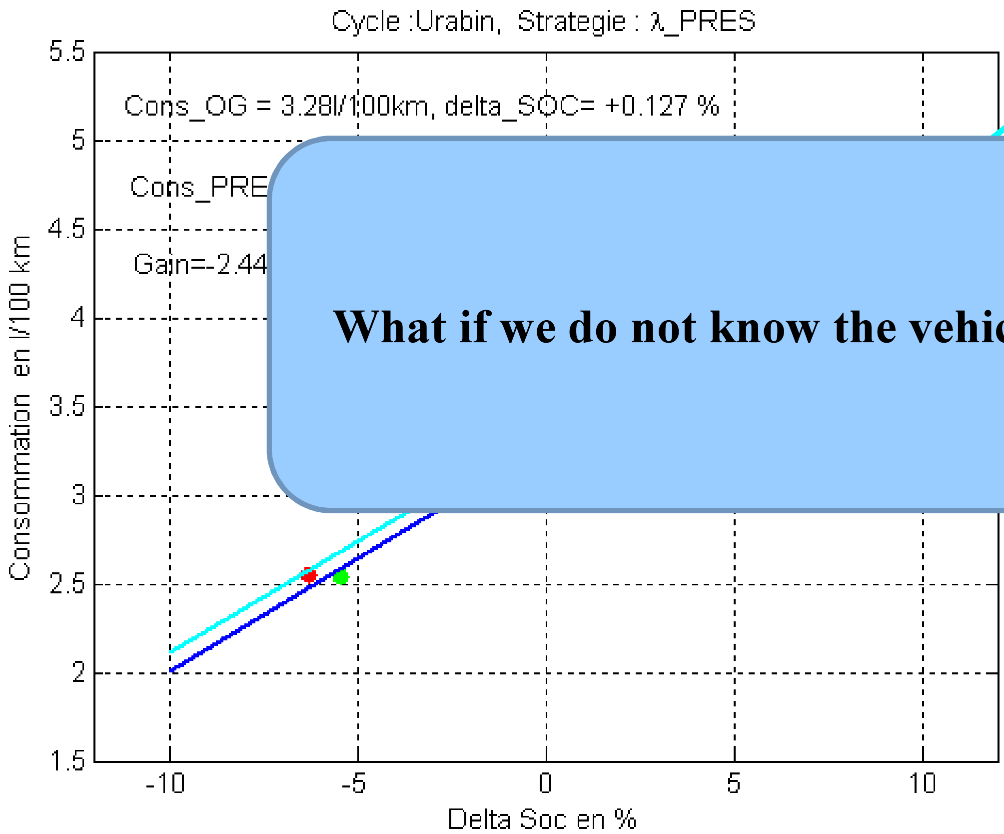


Vehicle on predefined routes

Experimental results:

- The algorithm can be executed in real time
- One of the experimental driving cycle is chose as reference
- A second driving cycle is used for the experiment

$x(N) - x(0) = -1{,}52\ \%$ *Fuel cons. = 3,16 l/100km*



λ_{PRES} control strategy:

…rol strategy
…l)

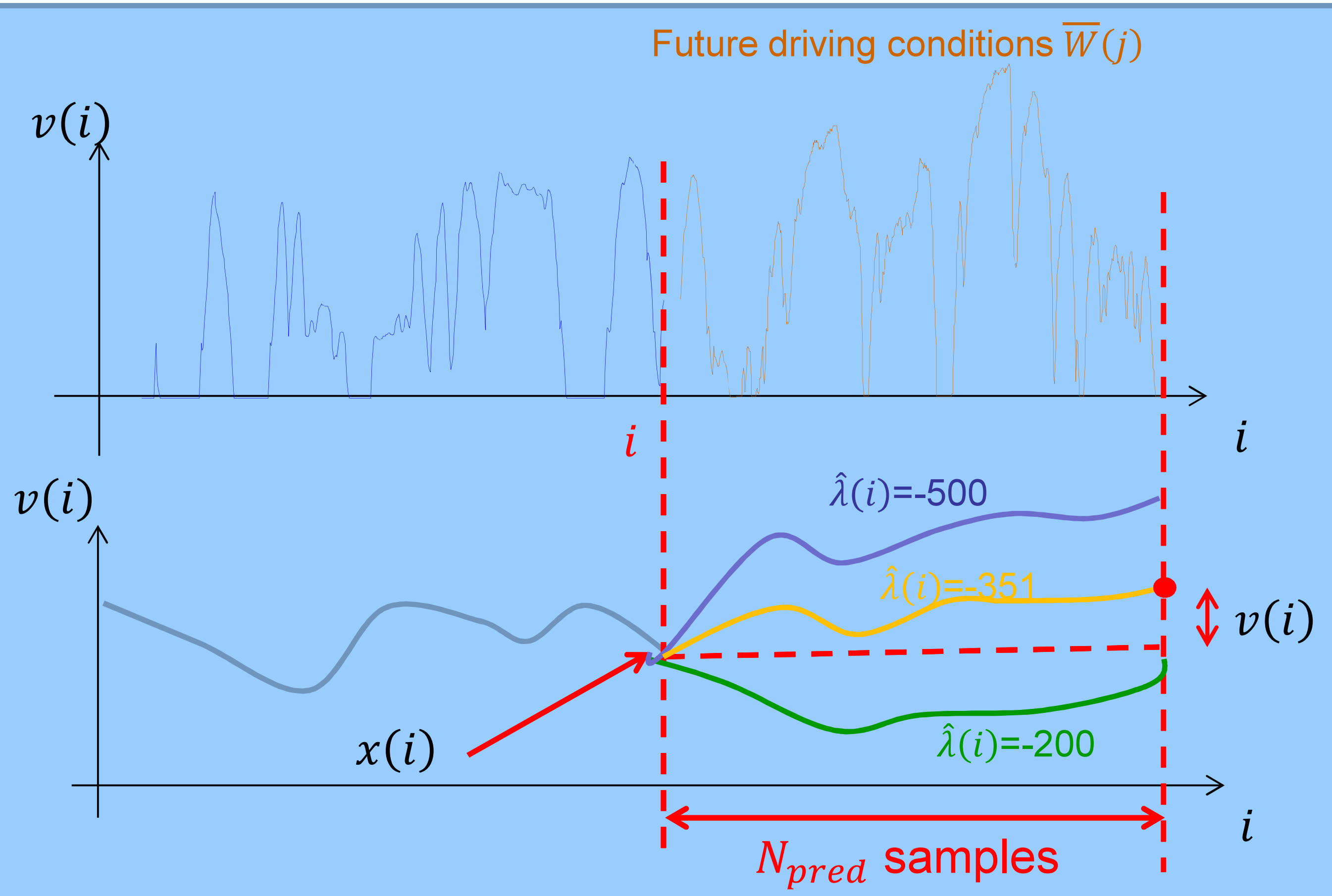
What if we do not know the vehicle itinerary ?

		Cons	Deviation cons
λ_{PRES}	0,12%	3,36l/100km	2,4%
FL(opti)	0,12%	3,28l/100km	/

Predictive control

Embed the optimization algorithm within a predictive framework:

- Future values are denoted with a bar
- Initial state value at the beginning of the horizon : $x(i)$ (known value)
- Fix an arbitrary expected final state value: $\bar{x}(i+N_{pred}) = x(i) + v(i)$



Predictive control

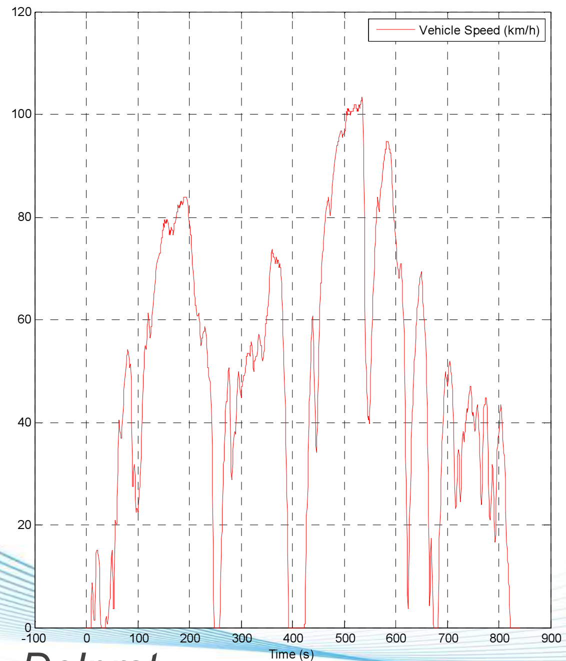
Let us remember….

State of charge at the end of the driving cycle

$$\bar{x}\left(i+N_{pred}\right)=x(i)+v(i)\Leftrightarrow\sum_{j\in J}M\left(\hat{\lambda}(i),\bar{W}(j)\right)\cdot s=v(i)$$

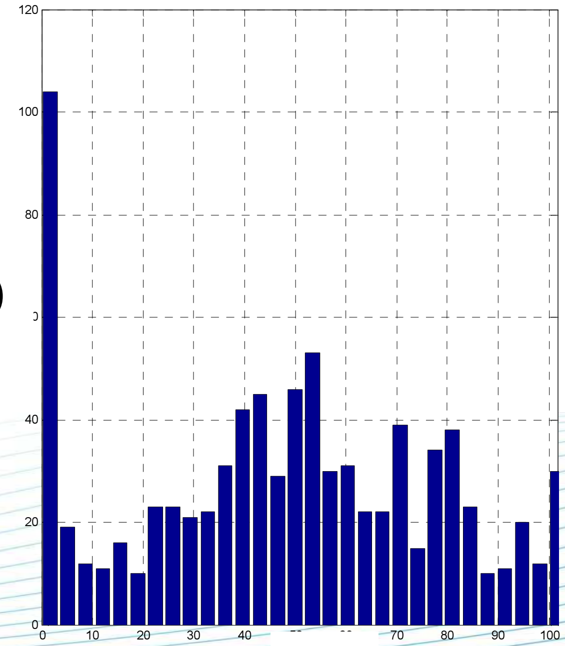
The sum is not ordered => Consider the frequency distribution of W:

$$\sum_{j\in J}M\left(\hat{\lambda}(i),\bar{W}(j)\right)\cdot s=\sum_{l=0}^{n_l-1}\mu(l)\cdot M\left(\hat{\lambda}(i),W_l\right)\cdot s$$



Space quantification

$\mu(l)$



W_l

Predictive control

The sum is not ordered => Consider the frequency distribution of W:

$$\sum_{j \in J} M\left(\hat{\lambda}(i), \overline{W}(j)\right) \cdot s = \sum_{l=0}^{n_l - 1} \mu(l) \cdot M\left(\hat{\lambda}(i), W_l\right) \cdot s$$

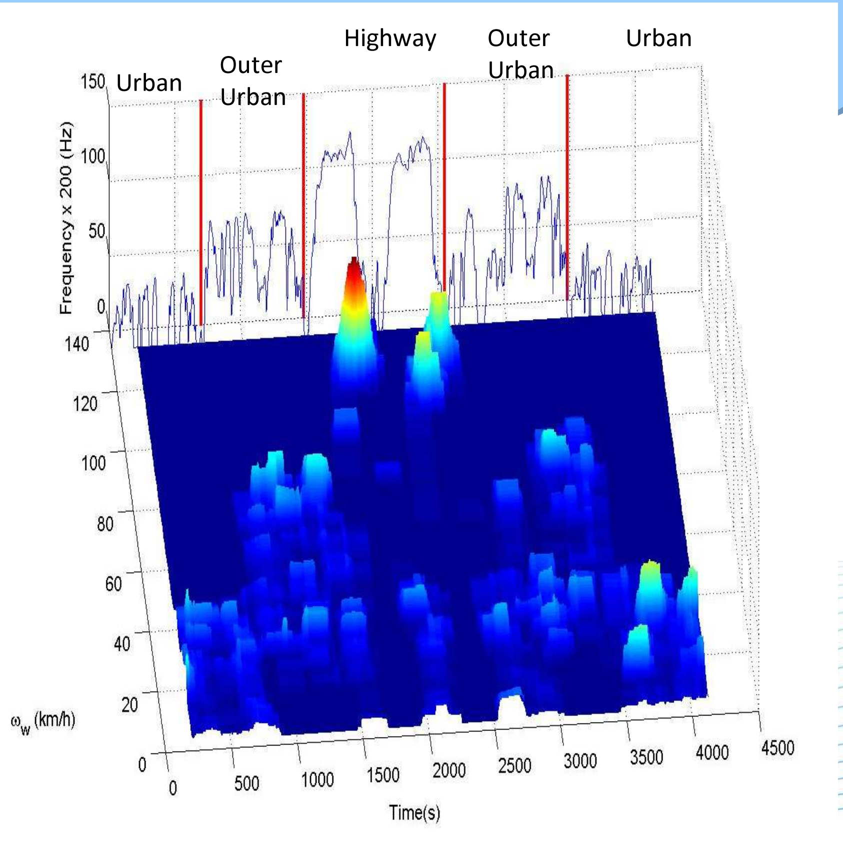
less computation if $N_{pred} \gg n_l$

- The frequency distribution of W is piecewise constant

$\overline{W}$ frequency distribution

Example: Vehicle speed frequency distribution over a 54 km long trip

Windows length : 200 s



Frequency distribution assumption

Assumption:

There exist a time window, such that the frequency distribution of $\overline{W}(i)$ over the future window is similar to the one obtained over the previous window

Predicted   Measured

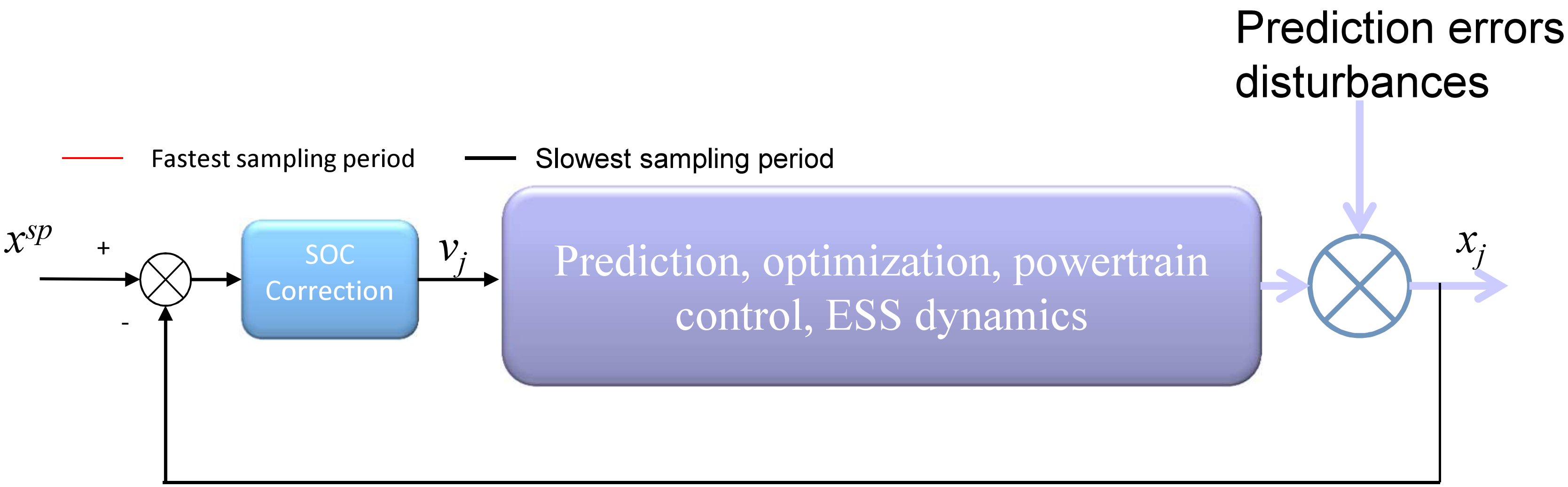
$$\underbrace{\sum_{i \in I^+} M\left(\lambda_j, \overline{W}(i)\right) \cdot s}_{FUTURE} \approx \underbrace{\sum_{i \in I^-} M\left(\lambda_j, W(i)\right) \cdot s}_{PAST}$$

$$I^- = \left[(j-1)N_{pred}, j \cdot N_{pred} - 1\right]$$

$$I^+ = \left[j \cdot N_{pred} ; (j+1) \cdot N_{pred} - 1\right]$$

A shooting algorithm is available for real time control

Predictive control



Soc Correction : adjust the SOC target according to disturbances

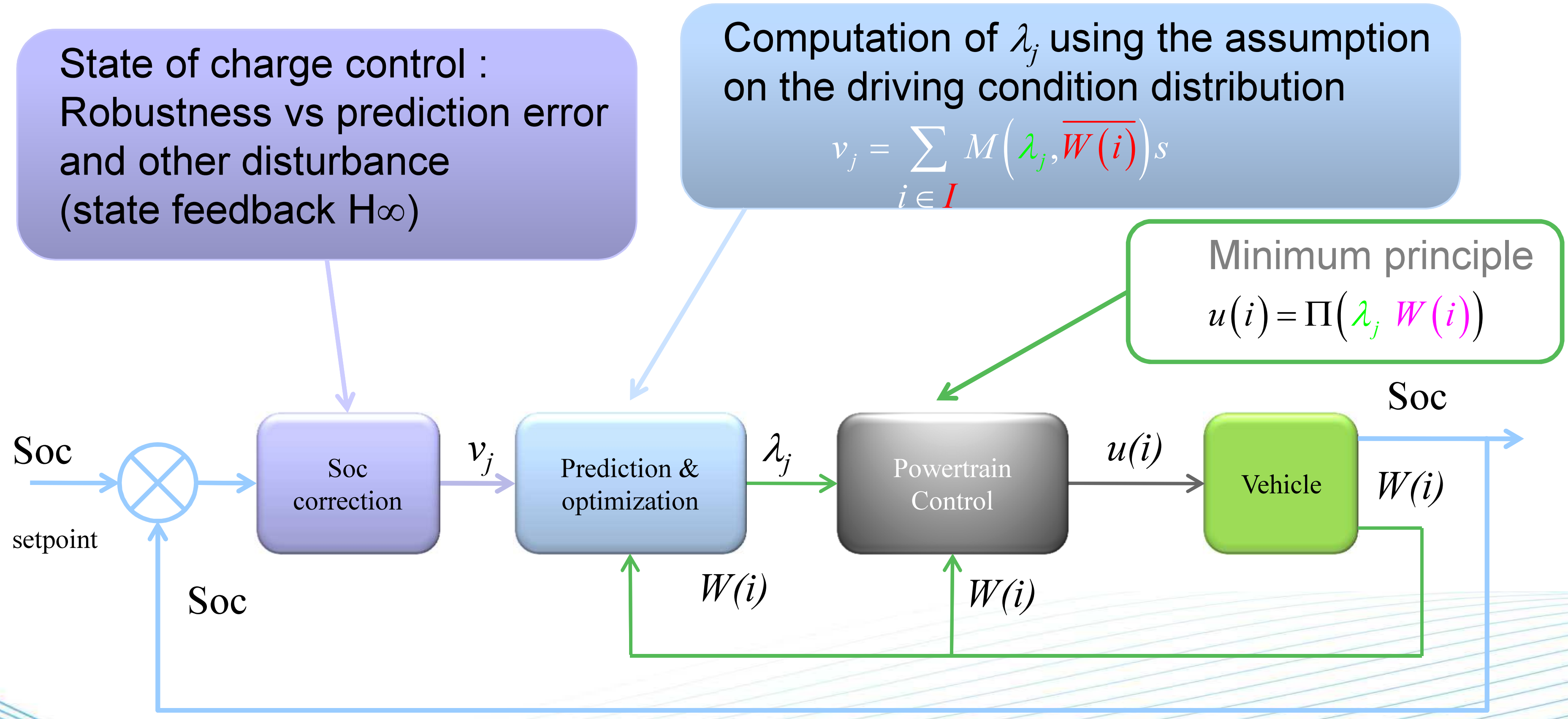
$$x_{j+1} = x_j + v_j + d_j + p_j$$

$\Rightarrow$ Linear controller insure stability
$\Rightarrow$ Input to State Stability apply

Best performances with respect to prediction errors:
=> Proportional K=1
Persistent disturbances rejection:
=> PI controller

Predictive control

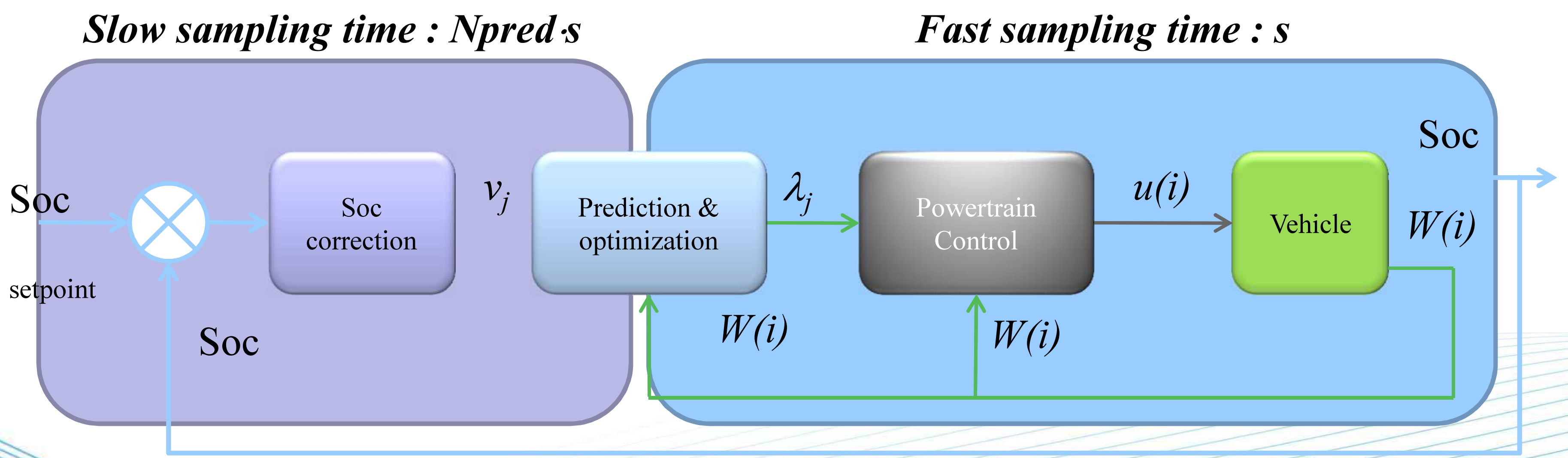
Outline of the λ_{PRED} algorithm:



Predictive control

Control strategy structure :

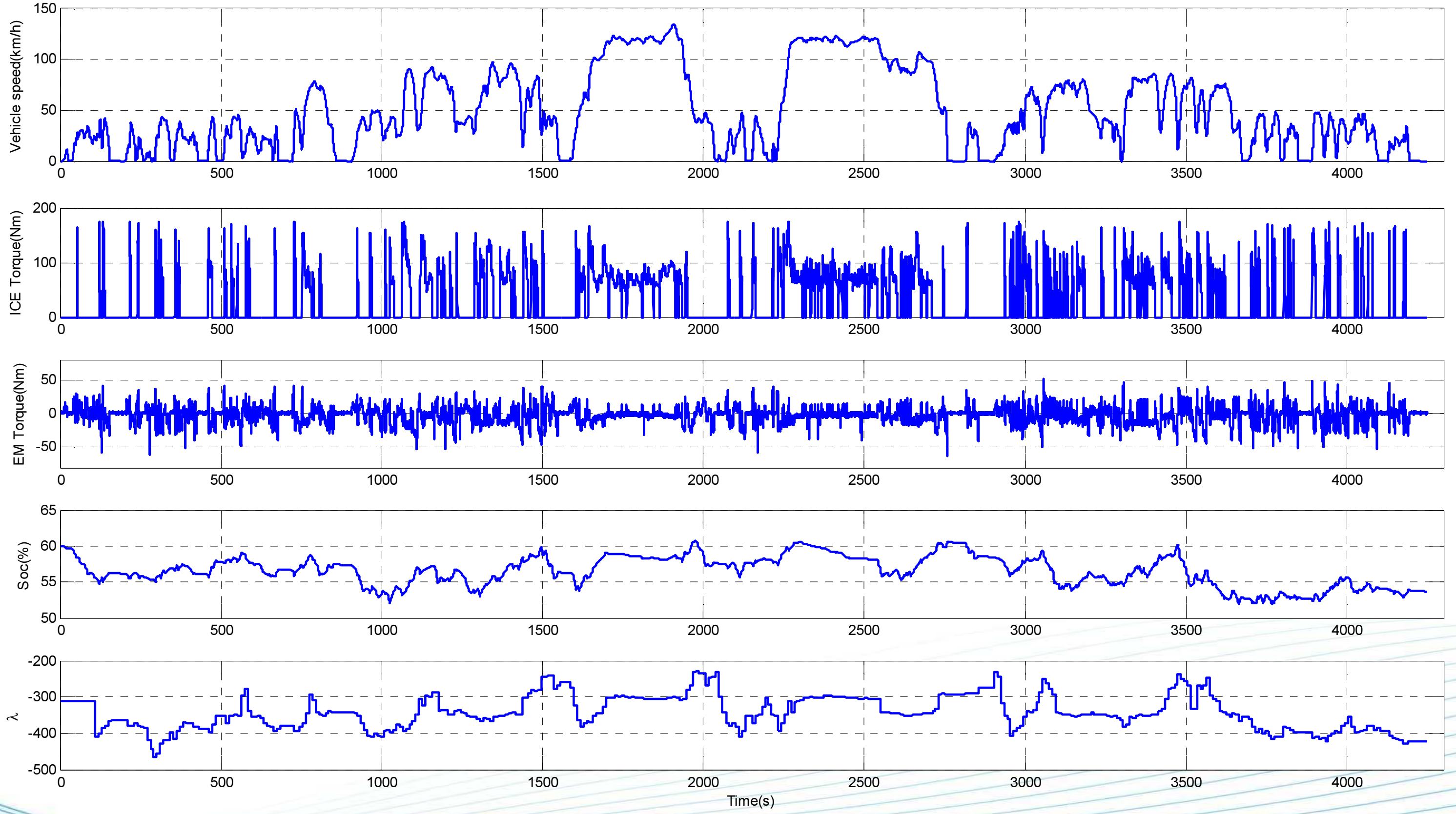
- Powertrain control : Fast update to cope with the dynamics of actual exogenous variables W(i)
- Soc correction, Prediction : slow sampling time & slow dynamics (in order of the battery SOC dynamics)



Predictive control

Experimental results : Mixed driving conditions

- 1h10min long
- Different driving condition
- SOC control is ok => Hypothesis holds in practice

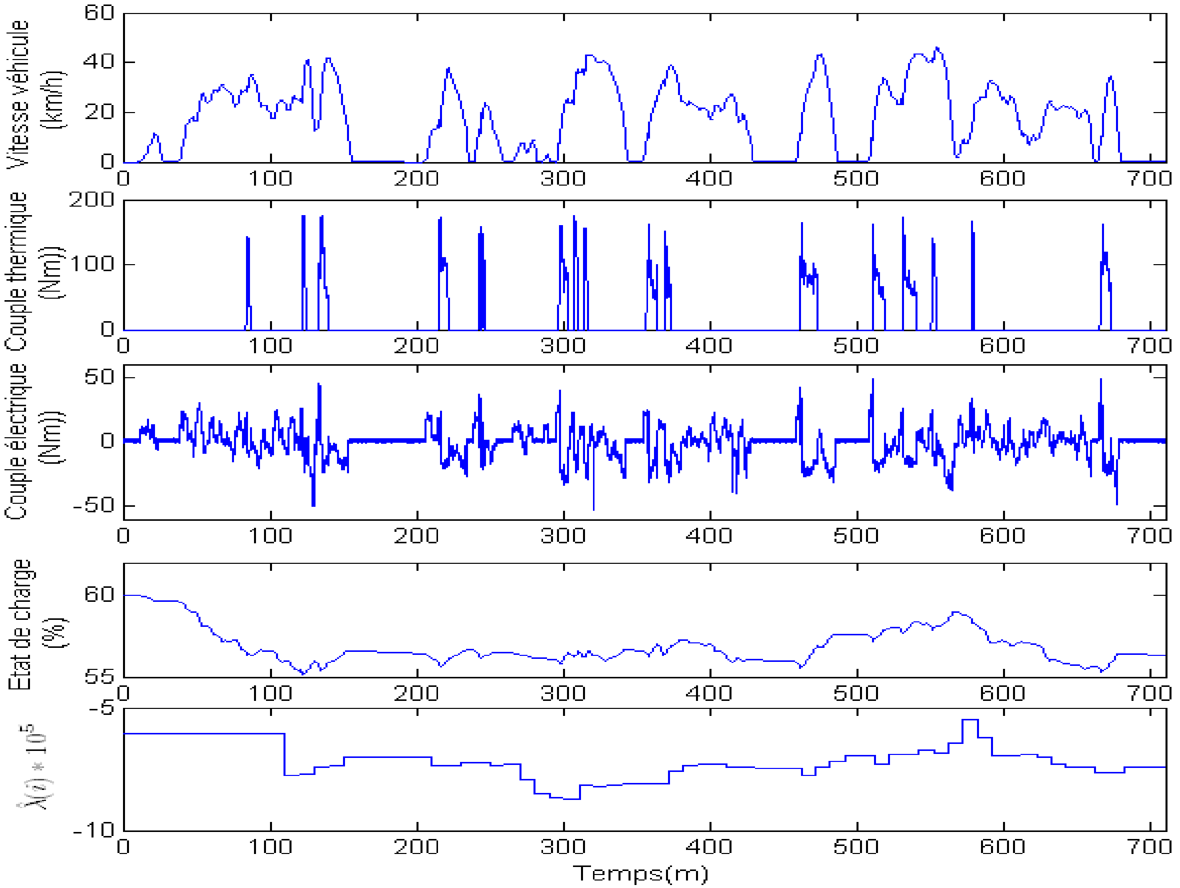


Predictive control

Experimental results : Outer Urban driving conditions

$$\begin{cases} s = 0{,}1s \\ \hat{\lambda}(0) = -6\times 10^{-5} \\ N_{PREV} = N_{PRED} = 1000,\ N_l = 100 \end{cases}$$

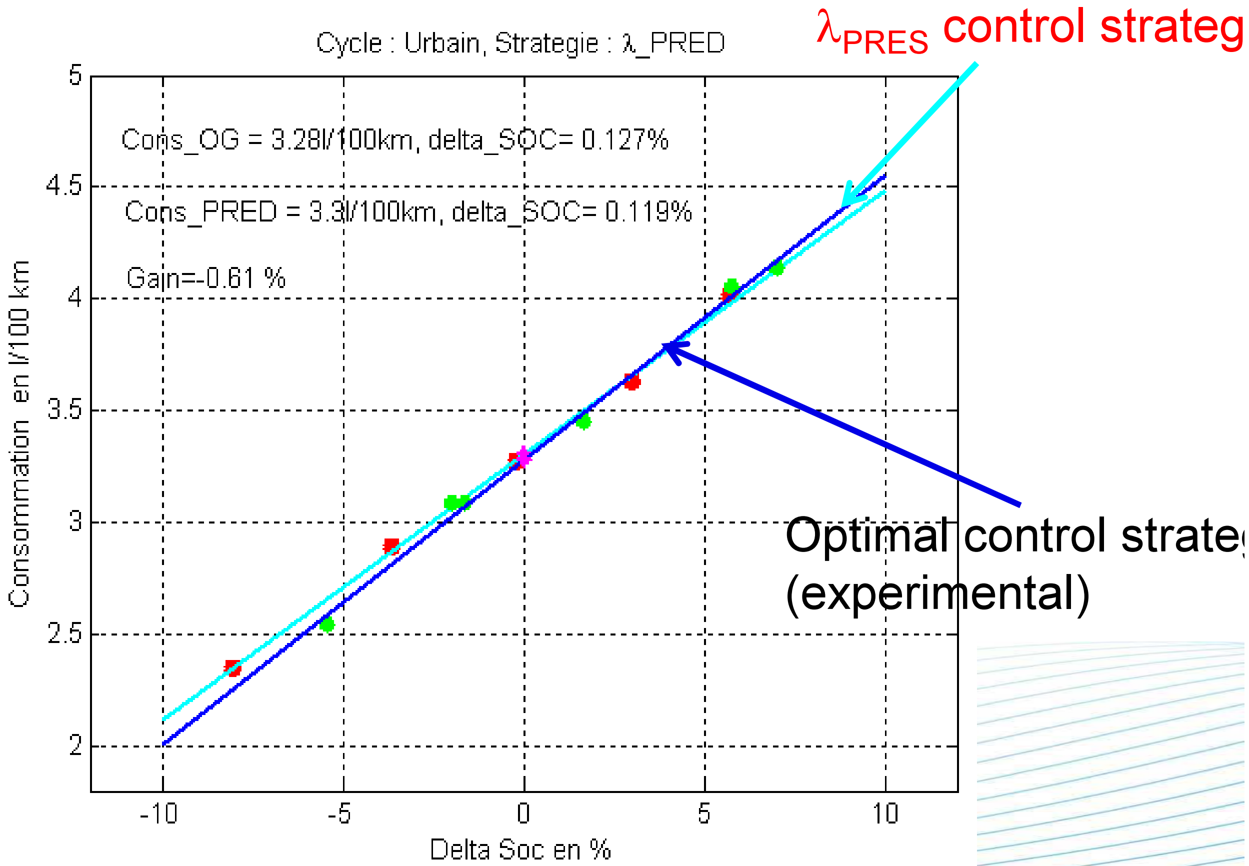
ΔX=-3,6%,Conso=2,89l/100km



Predictive control

Experimental results : Outer Urban driving conditions

$$\begin{cases} s = 0{,}1\,s \quad x(0) \approx 60\% \\ \hat{\lambda}(0) = -6 \times 10^{-5} \\ N_{PREV} = N_{PRED} = 1000,\ N_l = 100 \end{cases}$$



	ΔSOC	Cons	Deviation cons
	0,12%	**3,30l/100km**	**0,61%**
FL(opti)	**0,12%**	**3,28l/100km**	**/**

Conclusion

Designing energy management algorithm for hybrid vehicle is a challenging task.

Several theoretical points have to be well understood.

Some of them are directly related to the choice of a particular model structure

- Sparse IC engine maps or EM maps => A lot of singular controls
- Convex model => analytical Hamiltonian minimization

Some of them are related to the powertrain design

- State constraints related with ESS size

But theory is not all, practicing is necessary :

- Driving cumfort
- Mode transitionning
- Noise
- Thermal management

Conclusion

Designing energy management algorithm for hybrid vehicle is a challenging task.

Several theoretical points have to be well understood.

Some of them are directly related to the choice of a particular model structure

- Sparse IC engine maps or EM maps => A lot of singular controls
- Convex model => analytical Hamiltonian minimization

Some of them are related to the powertrain design

- State constraints related with ESS size

But theory is not all, practicing is necessary :

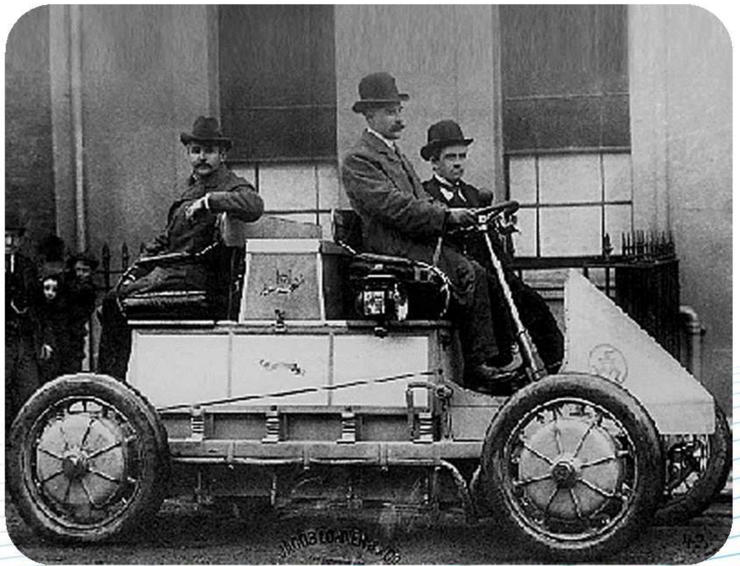
- Driving comfort
- Mode transitioning
- Noise
- Thermal management

Conclusion

Thank you for your attention

Theory is when one knows everything but nothing works.

Practice is when everything works but nobody knows why.



Contact:

S. Delprat
University of Valenciennes et du Hainaut Cambrésis
LAMIH UMR 8201
France
email : sebastien.delprat@univ-valenciennes.fr