GREMO

NAGOYA UNIVERSITY

BORDEAUX 2015

22nd

ITS World Congress

Bordeaux, France

5 to 9 October

2015

Behavior personalized Adaptive Cruise Control using Probability-Weighted ARX model

Thomas WILHELEM, Hiroyuki OKUDA

Blaine LEVEDAHL, Tatsuya SUZUKI

Tetsunori HARAGUCHI

TOWARDS INTELLIGENT MOBILITY

Better use of space

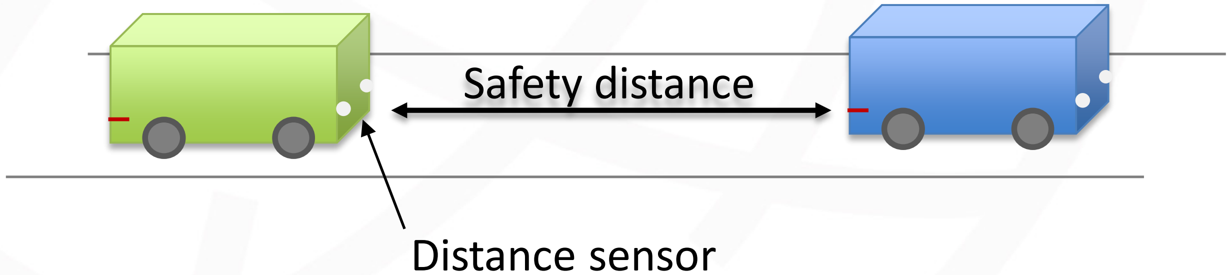
Introduction

Goal of the adaptive cruise control:

- Adjust the vehicle velocity to maintain a safety distance with the lead vehicle. Relieve the driver.

How adjust the velocity:

- By sensing the lead vehicle (with a Radar or Lidar), and correcting the vehicle velocity.



Different approaches exist to evaluate the safety distance

Introduction

Most adaptive cruise controls algorithms are derivated from two different microscopic traffic flow model types:

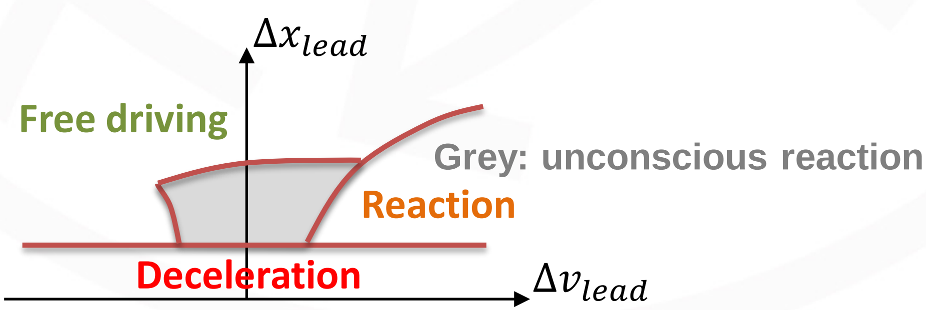
- Collision avoidance models (Gipps model …)

  The model applies the lowest of two possible accelerations:
  - Free driving acceleration $\gamma(\Delta v_0)$
  - Safe braking acceleration $\gamma(\Delta x_{lead}\ \&\ v)$

    with Δv_0 the relative velocity to a fixed velocity
    and Δx_{lead} the relative distance to the leading vehicle

- Psycho-physical models (Wiedemann model …)

  The acceleration of the vehicle is based on different possible formula.
  The selection of the formula depends on perception thresholds:



Not designed for simple user personalization

Table of content

Show the functioning of behavior personalized adaptive cruise control

1. Vehicle following models

SUZUKI LAB.

Microscopic flow model

- Linear models / Stimulus-response
  Output proportional to inputs
  e.g. GHR model, Helly models
- Collision avoidance / Safety distance
  Minimize a risk factor
  e.g. Gipps model, IDM
- Phycho-physical / Action point
  Human identified thresholds based model (decisions)
  e.g. Wiedemann model

Machine learning models

- Linear control theory
  Linear operation on the inputs
  e.g. PID, ARX *Too basic*
- Nonlinear / Stochastic
  Neural networks
  Hidden Markov models *Too complex*
- **Hybrid systems**
  Different driving modes (decision)
  Linear modes
  PWARX, **PrARX** (soft modes transitions)

Simplicity of linear control theory, enhanced by multiple modes reproducing the driver mental states

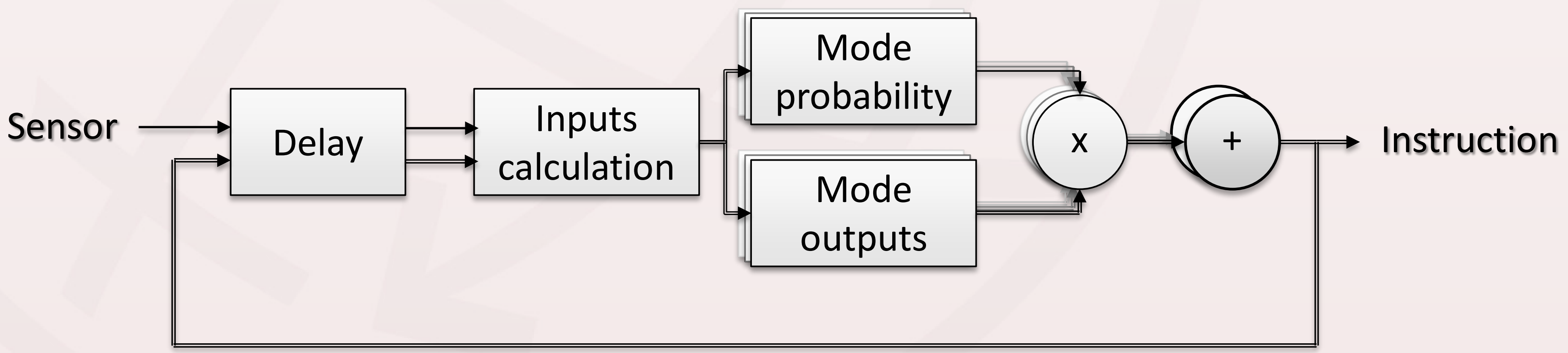
1. Vehicle following models

PrARX: ARX based model, probability weighted mode transition

Modeling the $(Vehicle + Driver)$ behavior

Supervised mode clustering [Acceleration ; Constant velocity ; Deceleration]

- Simple understanding of the modes parameters
- Sophisticated dynamics reproduction



Visualization of a 3 modes PrARX model

1. Vehicle following models

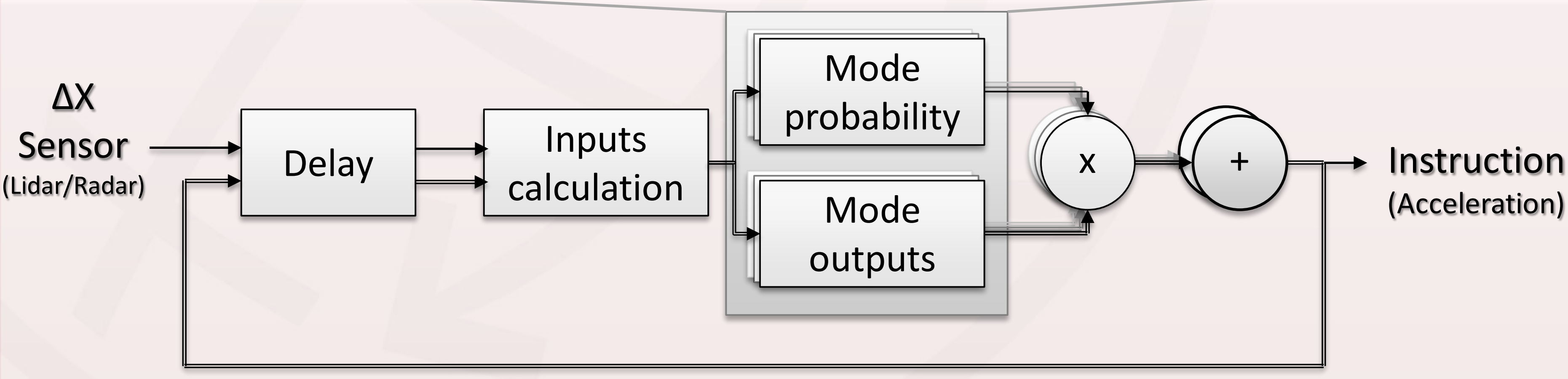
PrARX. Modeling the $(Vehicle + Driver)$ personalized behavior

Driver model inputs at $(t - \tau)$:
- **Vehicle acceleration**
- **Range** (Relative distance)
- **Range-rate** (Relative velocity)
- **Headway time**

Vehicle following model "PrARX"

Driver model output at t:
- **Vehicle acceleration**

τ expresses the driver reaction time



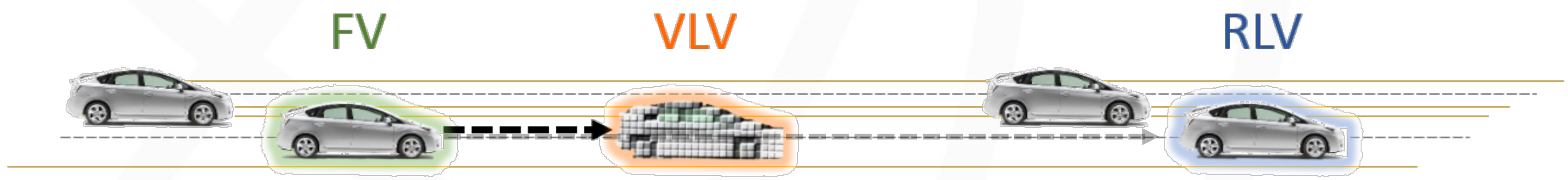
Visualization of a 3 modes PrARX model

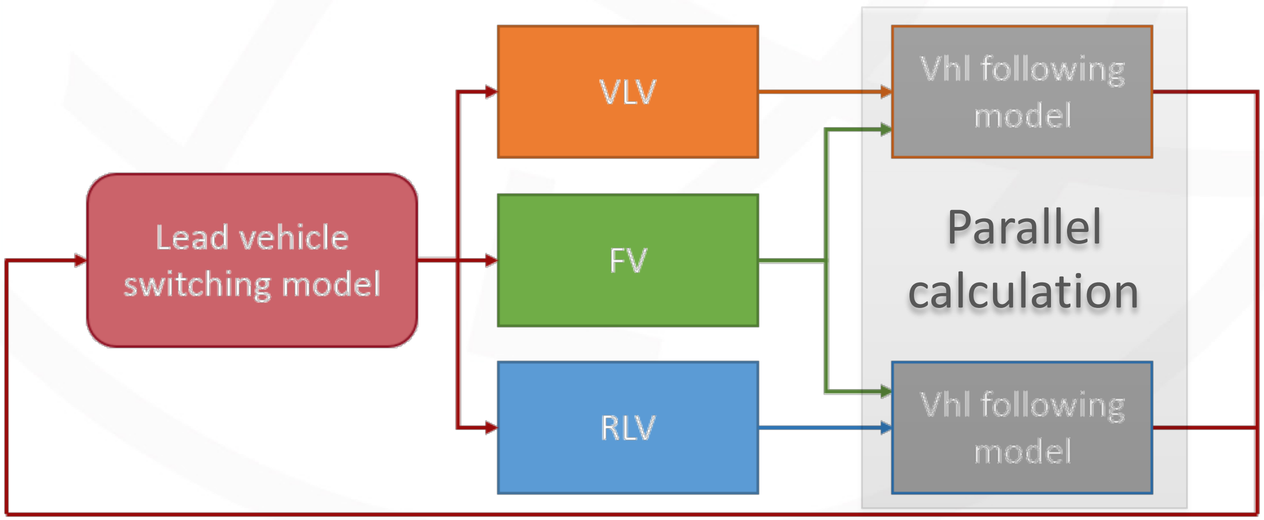
2. Application to A.C.C.

Application to Adaptive Cruise control: Vlv-ACC

Problem: What input when there is no leading vehicle?

➢ Introduction of the **Virtual Leading Vehicle (VLV)** concept





FV: Following vehicle (driven vehicle)

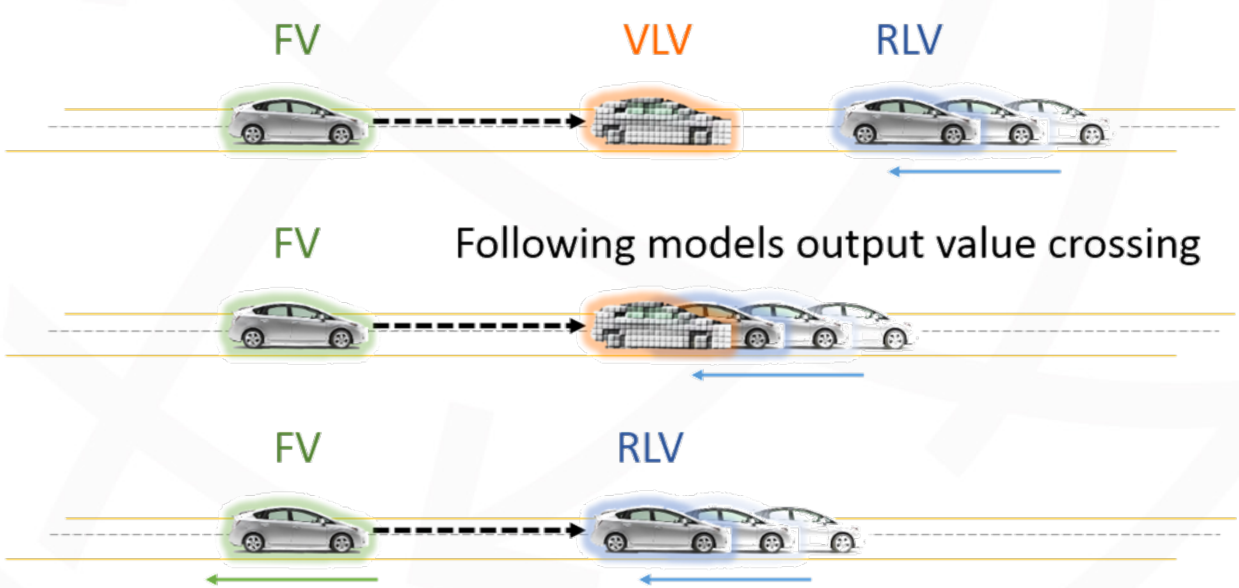
VLV: Virtual leading vehicle

RLV: Real leading vehicle

2. Application to A.C.C.

Vlv-ACC: **Switching cases**

1/4 "Soft RLV in"



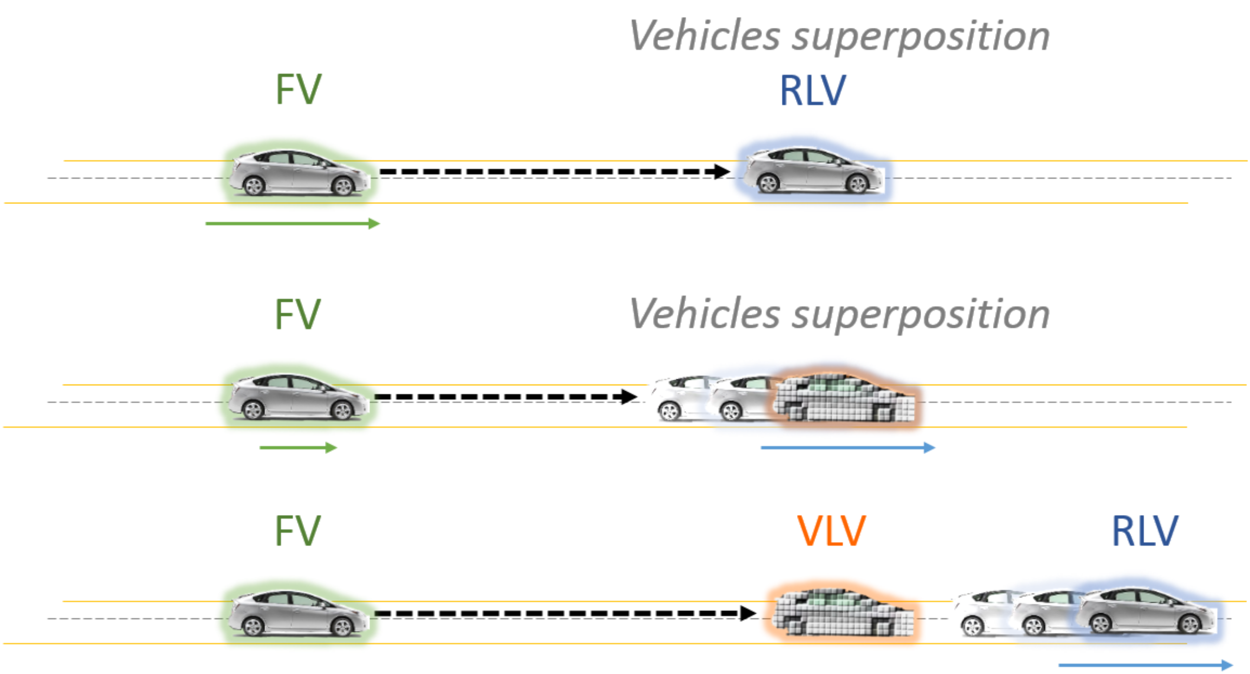
Following models outputs (acceleration)	
VLV	RLV
0	Positive
0	0
- - -	Negative

FV: Following vehicle (driven vehicle)
VLV: Virtual leading vehicle
RLV: Real leading vehicle

2. Application to A.C.C.

Vlv-ACC: **Switching cases**

2/4 "Soft RLV out"



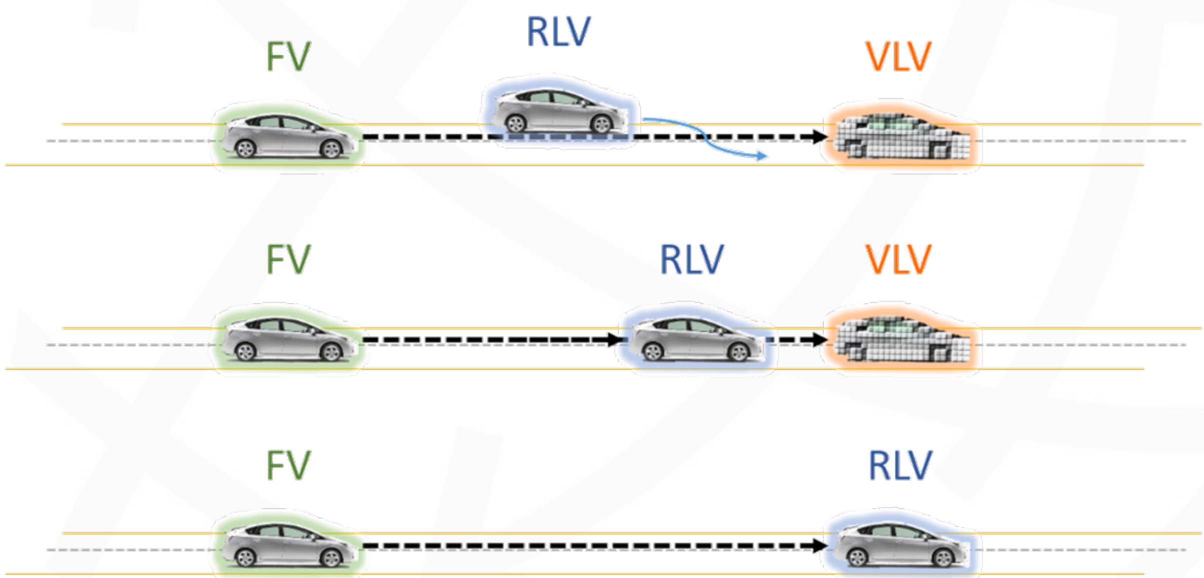
Leading vehicle velocity	
VLV	RLV
Same as RLV	Lower than ACC defined
ACC defined velocity	ACC defined velocity
ACC defined velocity	Higher than ACC defined

FV: Following vehicle (driven vehicle)
VLV: Virtual leading vehicle
RLV: Real leading vehicle

2. Application to A.C.C.

Vlv-ACC: **Switching cases**

3/4 "Lane cut RLV"



Following models outputs			
T	VLV	RLV	FV
$T < T_0$	a_x	- - -	a_x
$T_0 < T < T_0 + \tau$	$a_x(T_0)$	a_y	$t * a_x(T_0) + (t-\tau) * a_y$
$T_0 + \tau < T$	- - -	a_y	a_y

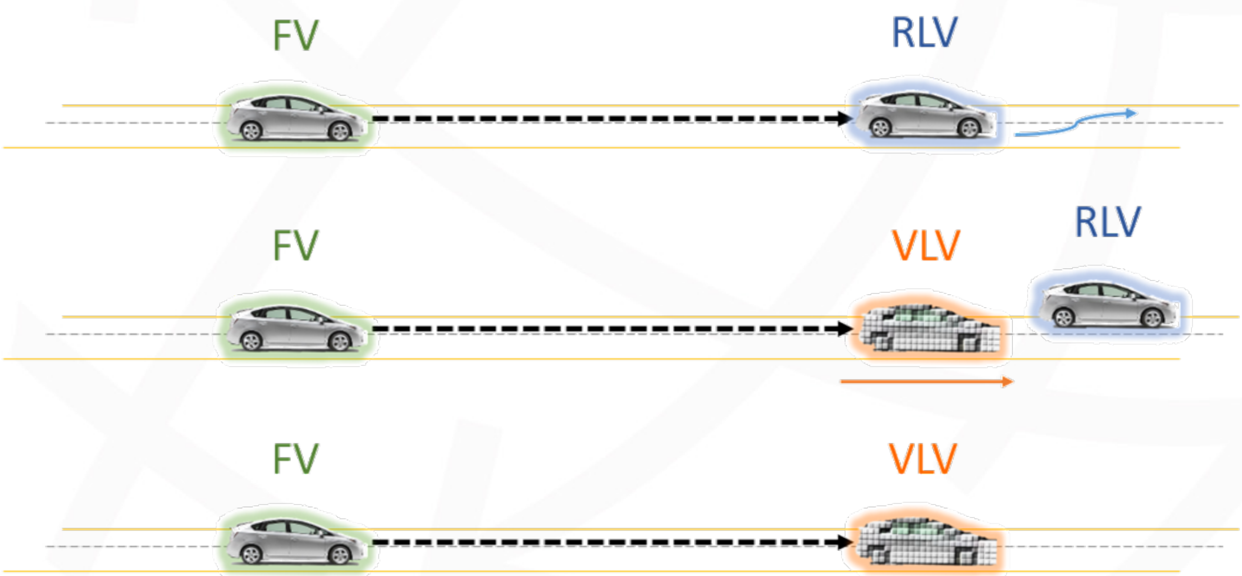
Timer based switching: τ

FV: Following vehicle (driven vehicle)
VLV: Virtual leading vehicle
RLV: Real leading vehicle

2. Application to A.C.C.

Vlv-ACC: **Switching cases**

4/4 “Lane exit RLV”



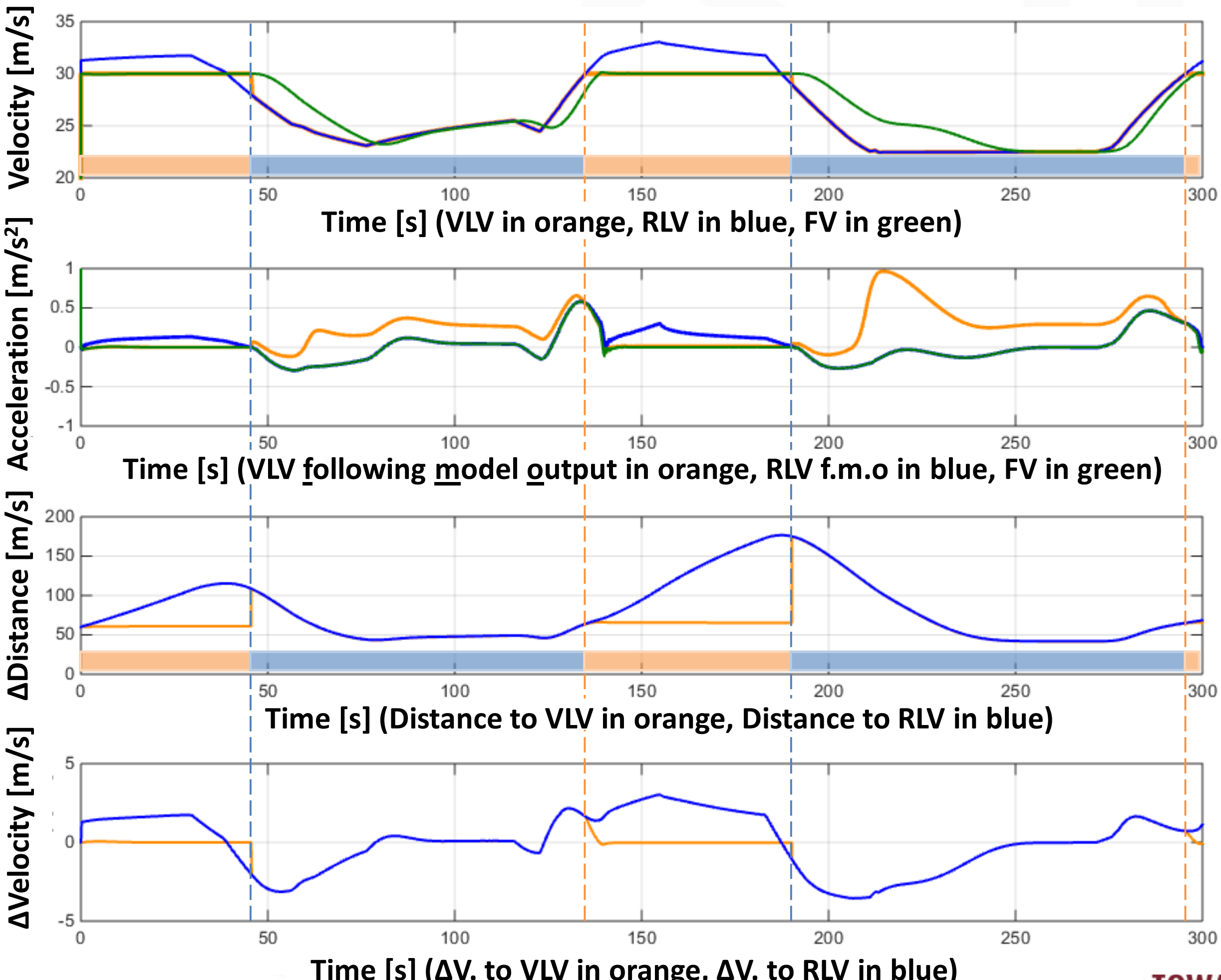
Vehicles accelerations		
Conditions	VLV	RLV
- - -	- - -	a_{RLV}
$V_{VLV} < V_{ACC}$	$a_{defined}$	- - -
$V_{VLV} = V_{ACC}$	0	- - -

Defined acceleration based switching

FV: Following vehicle (driven vehicle)
VLV: Virtual leading vehicle
RLV: Real leading vehicle

3. Results and comparison

Results: Vlv-ACC with PrARX model. <u>Soft transitions</u>



Target velocity: 30m/s

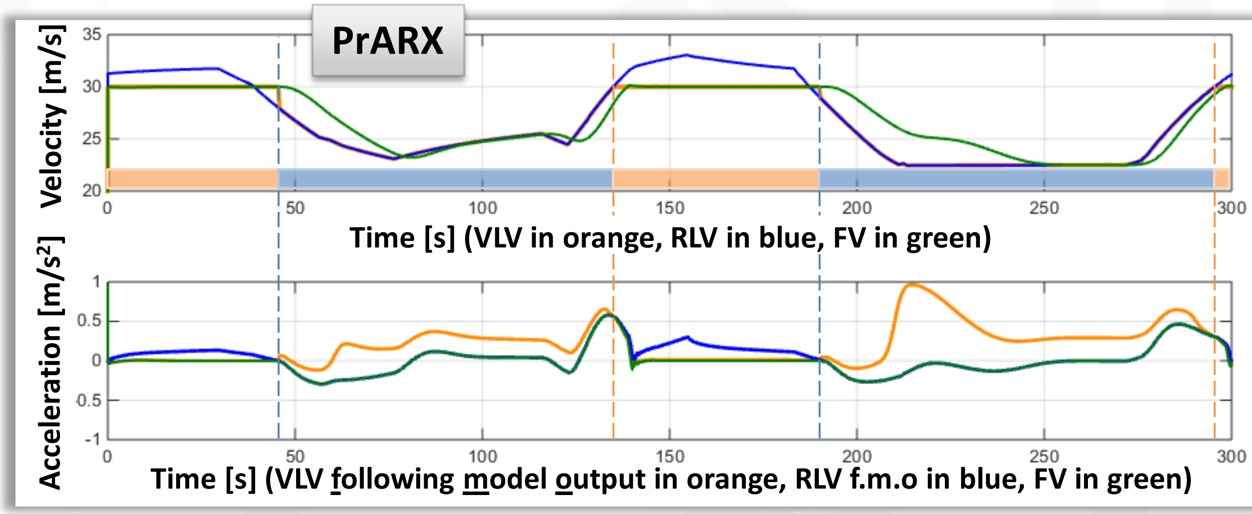
Following the
Virtual leading vehicle

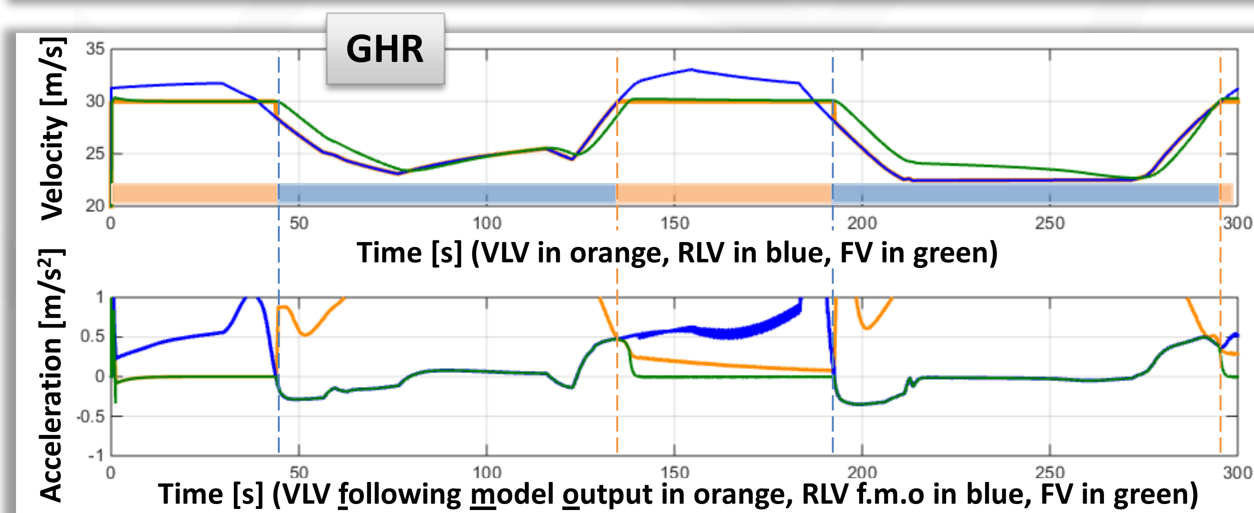
Following the
Real leading vehicle

Fig: Illustration of the vehicles

3. Results and comparison

Comparison: Vlv-ACC PrARX & Vlv-ACC GHR





Target velocity: 30m/s

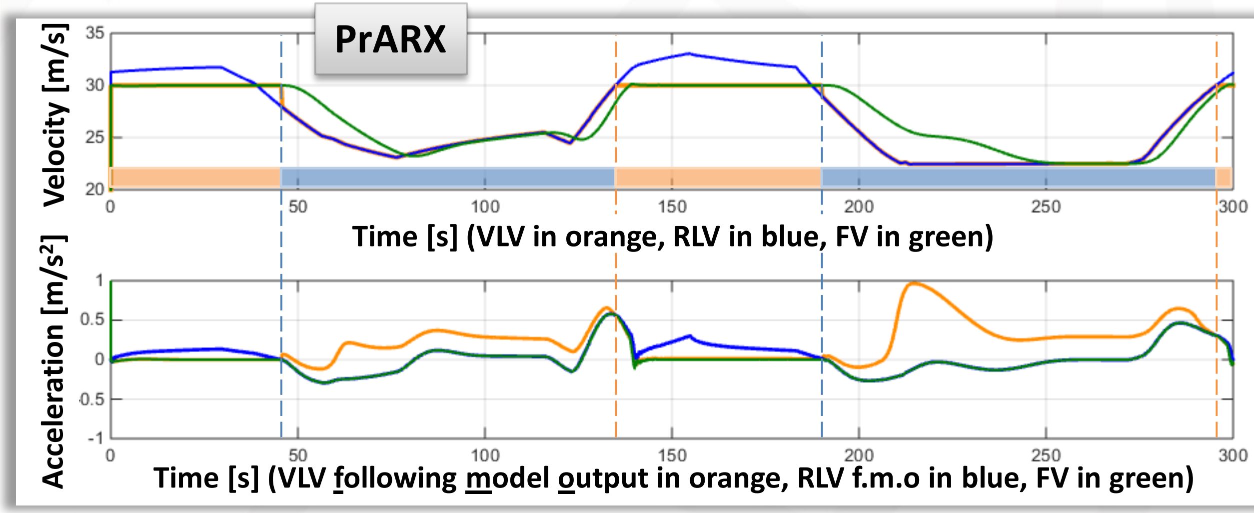
Following the Virtual leading vehicle

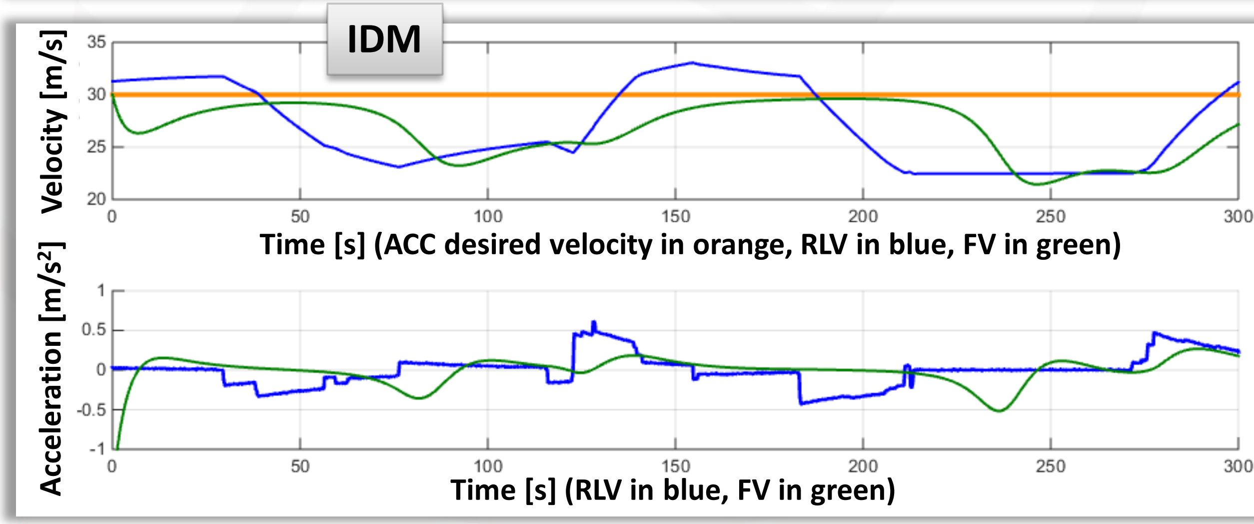
Following the Real leading vehicle

Fig: Illustration of the vehicles

3. Results and comparison

Comparison: Vlv-ACC PrARX & IDM





Target velocity: 30m/s

Following the
Virtual leading vehicle

Following the
Real leading vehicle

Fig: Illustration of the vehicles

Conclusion

We introduced:

- The PrARX machine learning vehicle following model
- The Vlv-ACC system, enabling to use vehicle following models to act as adaptive cruise control systems

We exposed:

- The inputs/outputs of the PrARX model
- The details of the Vlv-ACC functioning
- The comparison with different driver and ACC models

Future work will be:

- Implement the Vlv-ACC in a traffic flow simulation environment
- Test the system in a real vehicle

Thank you for listening

Do you have questions?