ME Control Lab

Research activities, related to

Control of autonomous vehicles on a smart road

Dr. Shai Arogeti

Department of Mechanical Engineering

Ben-Gurion University of the Negev

Intelligent Transportation Systems - Motivation

- Intelligent Transportation Systems (ITS) include a wide range of technologies that are aimed **at improving transportation systems**.

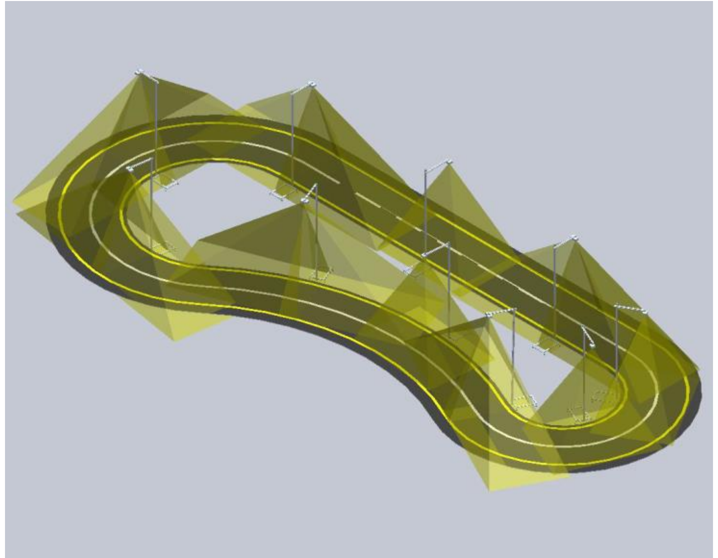
- Main goals include improvement in, **traffic efficiency**, **energy consumption**, **safety** (by removing the human factor) and **comfort** (by releasing drivers from tedious tasks such as a long drive).

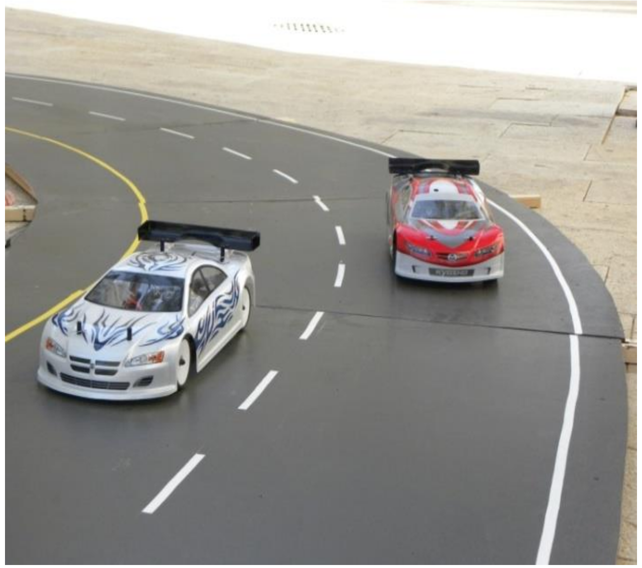
- Intelligent transportation systems **motivates** development of **efficient control algorithms for a group of unmanned autonomous vehicles.**

- **New concepts of smart roads** should be developed, which allows **autonomous vehicles and manually driven vehicles** on the same road (interaction, coordination, etc.)

Smart road – Test bed

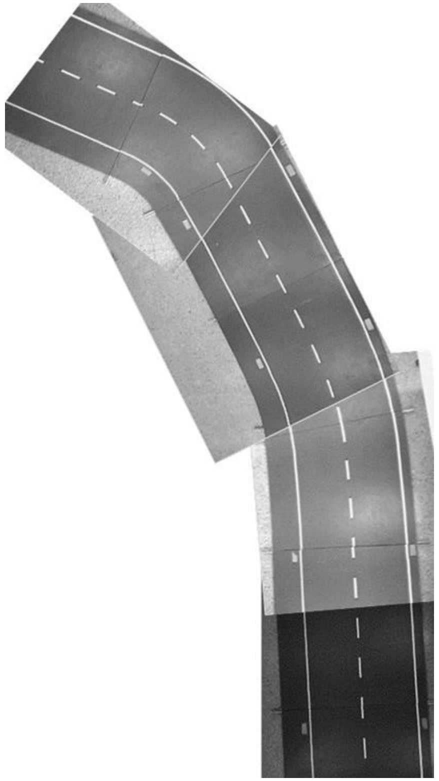
We are developing a concept of a smart road to allow autonomous driving of “almost” standard vehicles on a highway. The traffic may include also manually driven vehicles. The system is mainly based on a distributed network of sensors and controllers, **which are a part of the road infrastructure**.



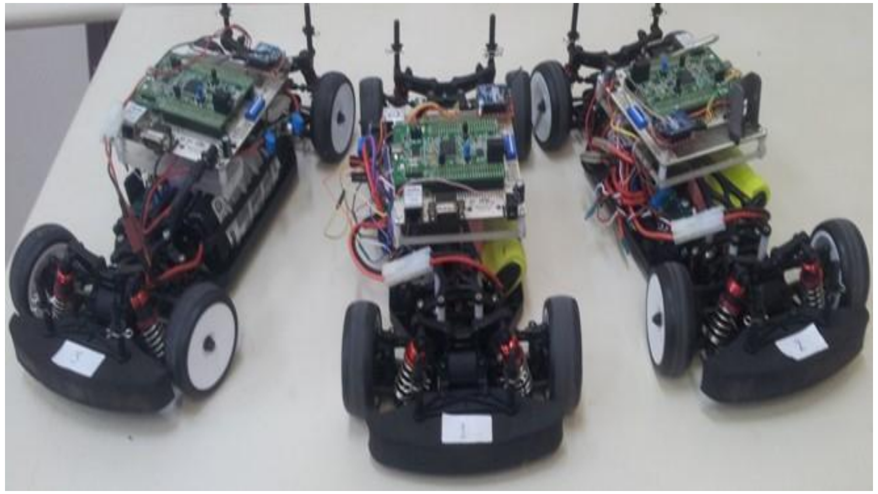


Smart road - Experiments

A road section, used as a Lab test-bed. This section is covered with four cameras.



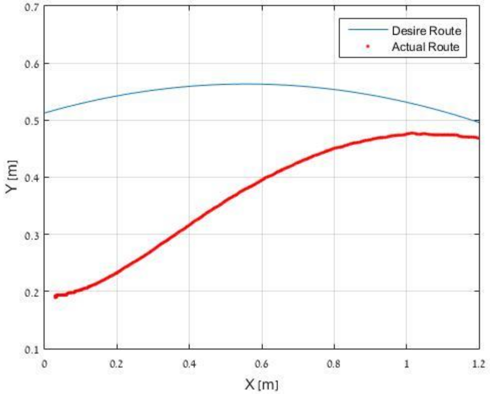
A safe environment for testing ITS algorithms



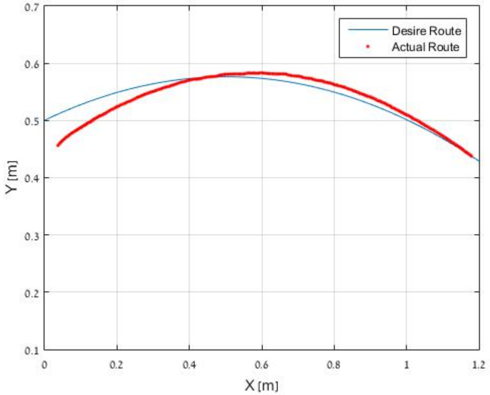
Smart road - Experiments

Experimental Results

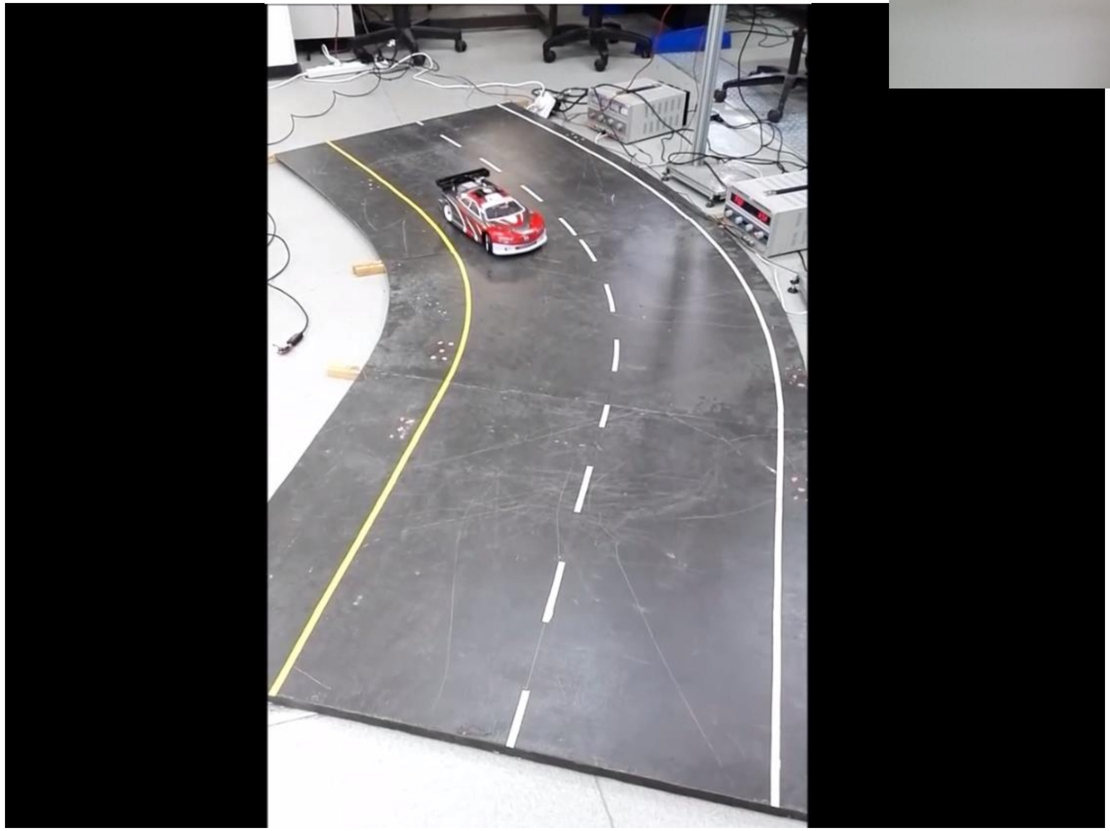
Trajectory (camera 1)



Trajectory (camera 2)



Raspberry Pi + Pi noir camera (for vision and control)



Driving commands are sent wirelessly

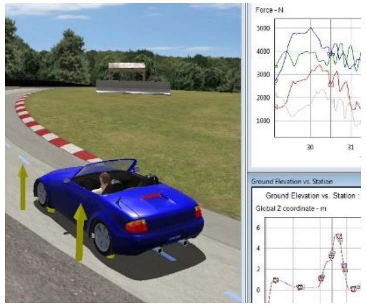
Smart road – CarSim Simulation

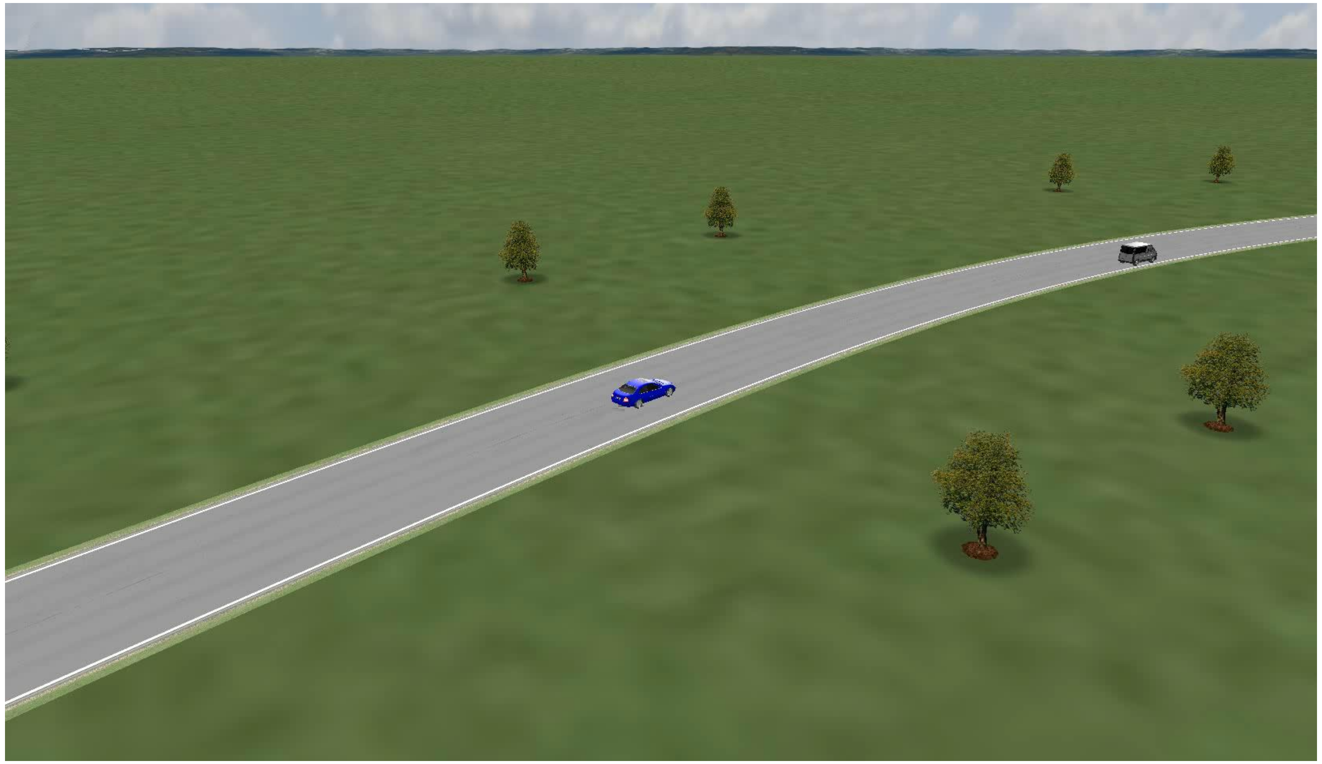
CarSim includes accurate, detailed, and efficient methods for simulating the performance of passenger vehicles.

CarSim has a standard interface to MATLAB/Simulink

Twenty years of real-world validation.

carSIM
MECHANICAL SIMULATION





CarSim Simulation

Stage 1 – Path following

Cruising speed

Stage 2 – Path following

Keeping front Safe distance (from vehicle ahead)

Autonomous vehicles – Development Trend

Three Generations

Control of a single vehicle

2000-2010

Path following

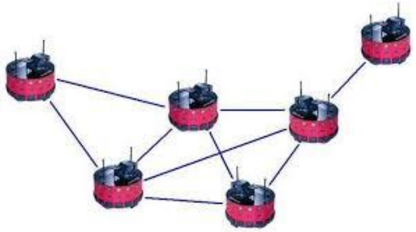
Collision avoidance

Control of a group of vehicle

2005-2015

Coordinated Path following

Formation, collision avoidance



Intelligent (and safe) control of a group of vehicle

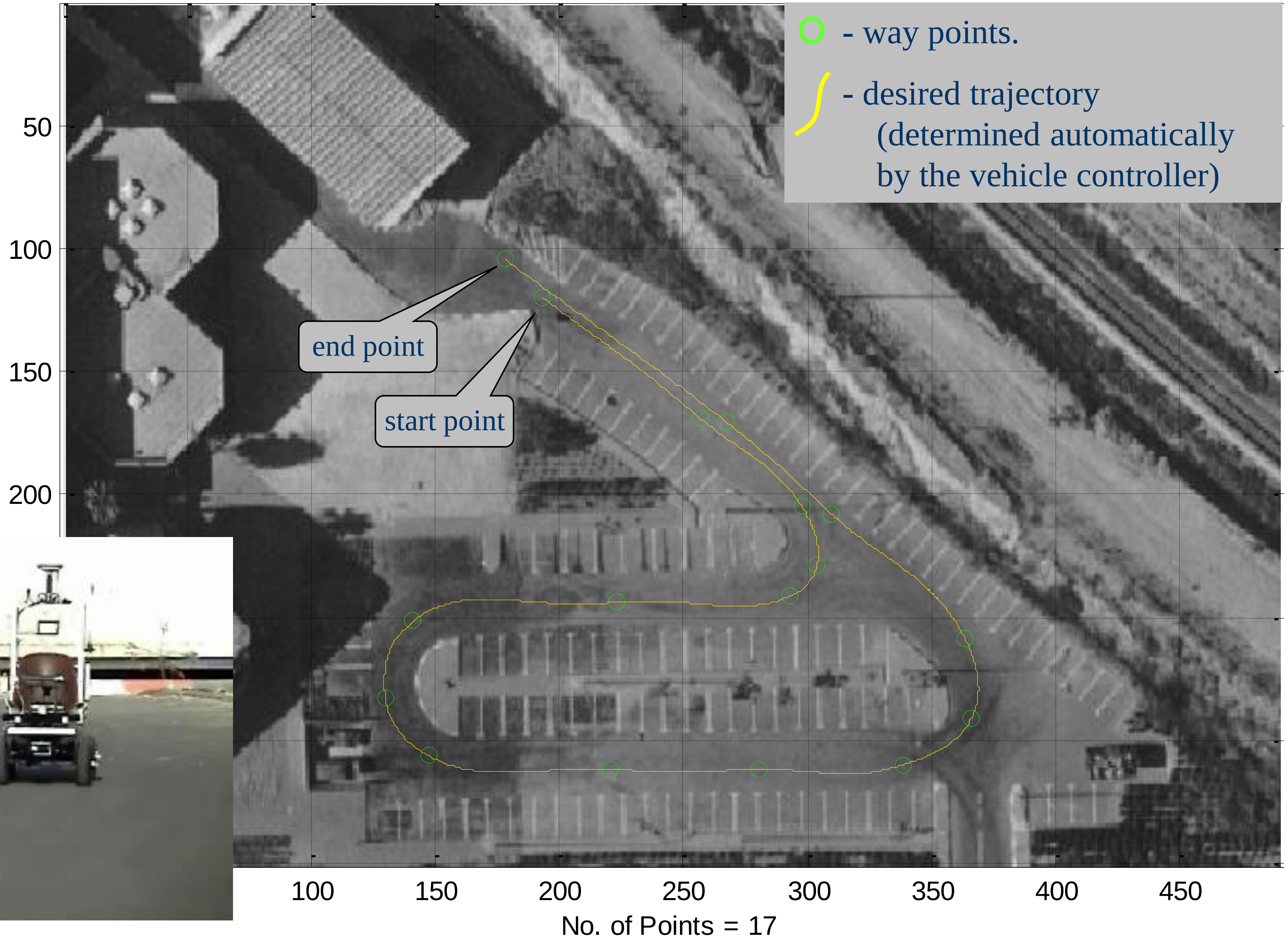
2015- . . . .

E.g., Use control tools to detect cyber attacks

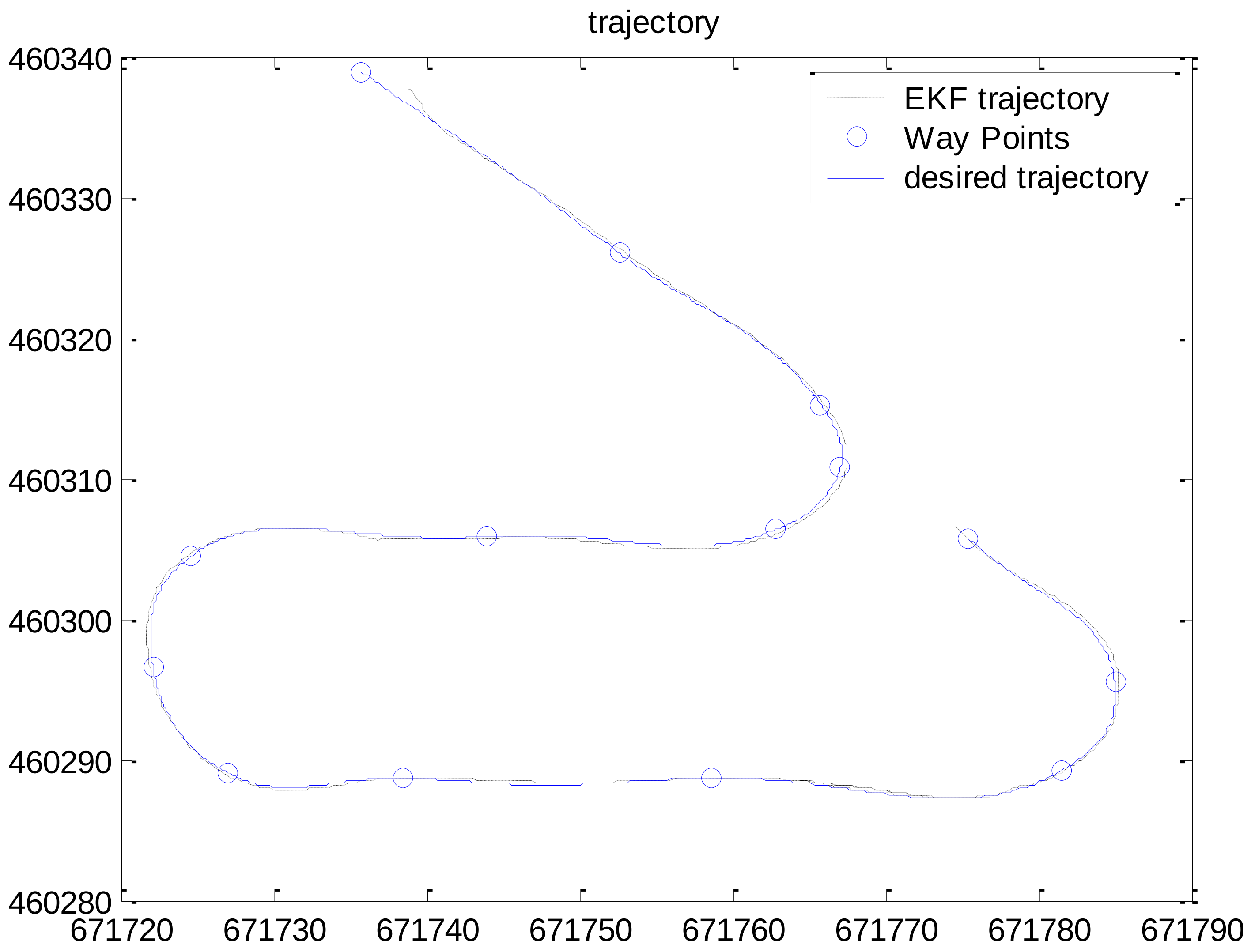
Intelligent decision making

A test bed – to try path following algorithms (year = 2004)

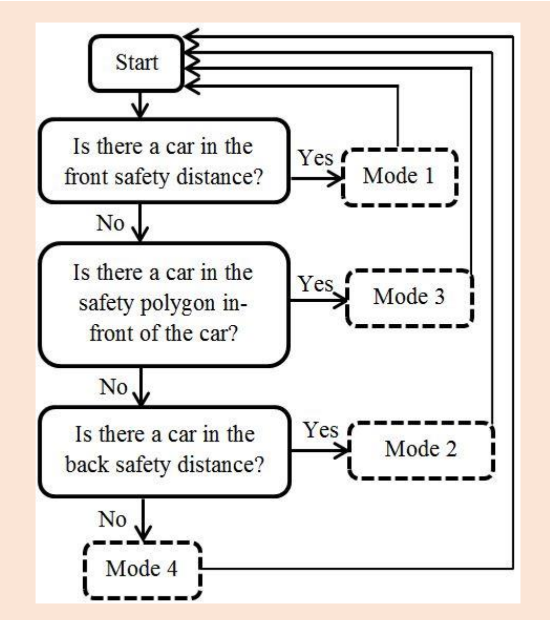
Experiment results:



Experiment results:

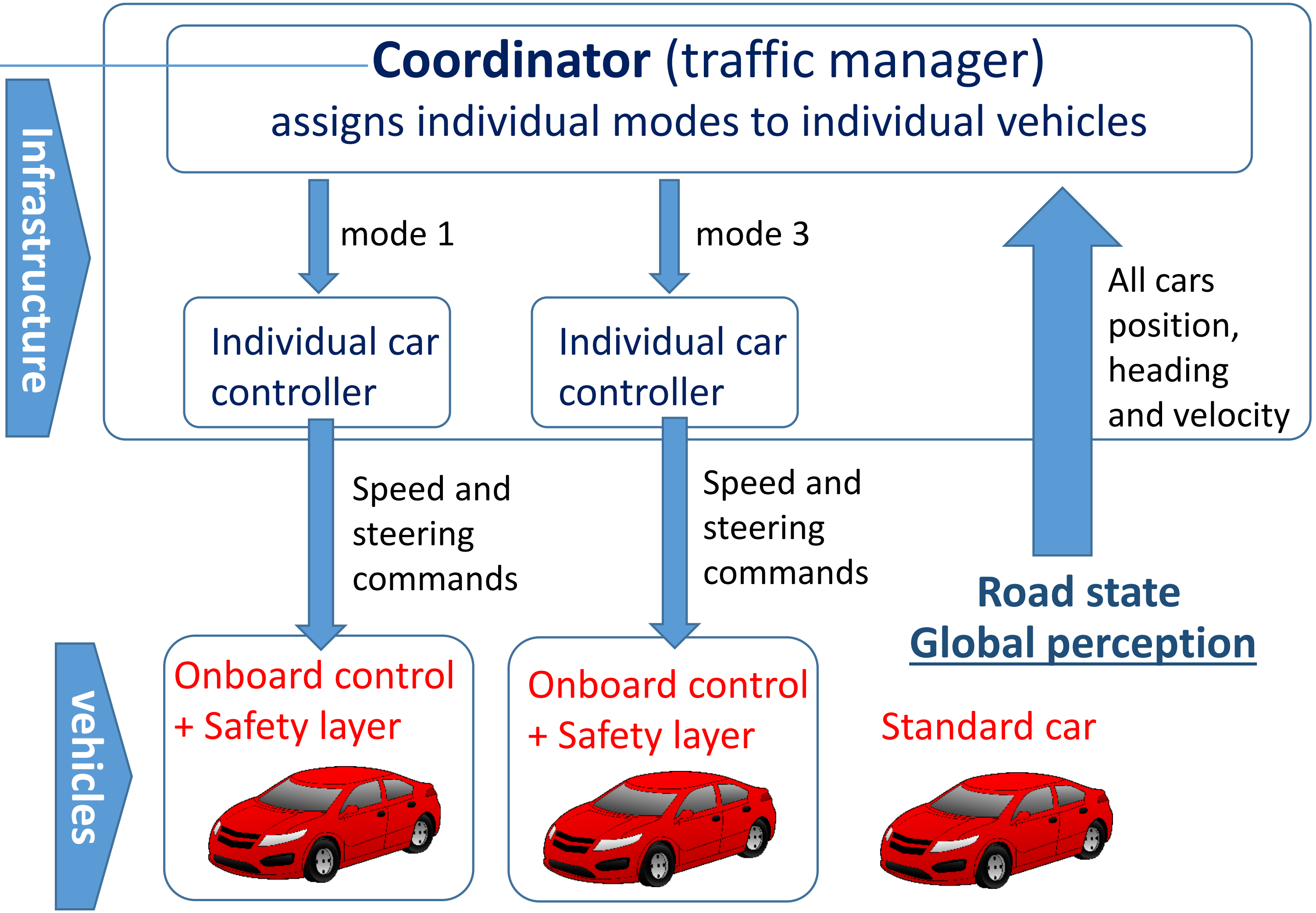


Smart road – Architecture



For example

A **multi-layer** control system **architecture**



This is a centralized solution and hence it is sensitive to cyber attacks

The local safety layer (not network connected) uses local perception (based on car sensors) to check the validity of the received commands.

It overwrites driving commands if these are not consistent with local information.

Smart road – Sensor and control redundancy for “fault” detection and accommodation

Each autonomous vehicle is equipped with two local autonomous-driving controllers,

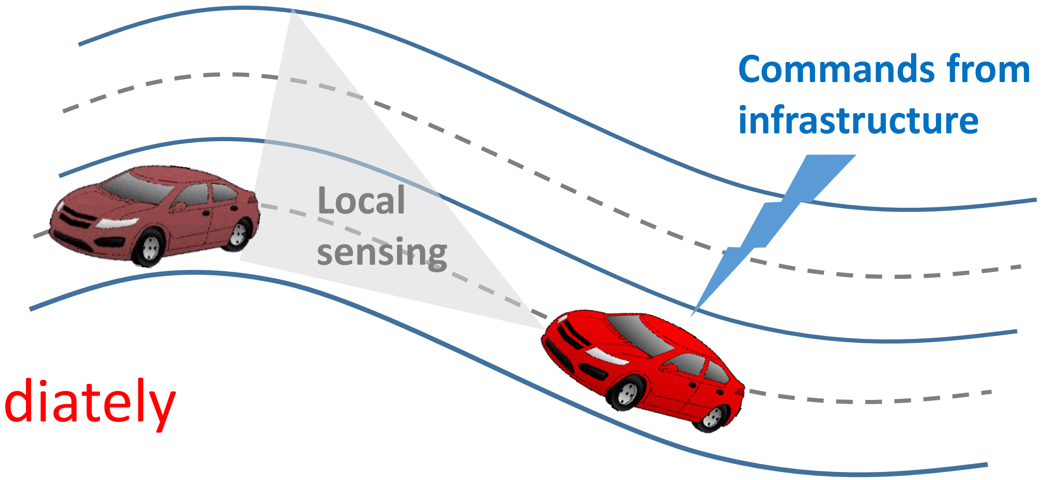
1. For path (road) following (→ steering command).
2. To keep safe distance from front vehicle (→ speed command)

These local controllers are for emergency situations only (they don’t use trafic perspective).

All Individual car controllers (car-infrastructure loop) are designed with a prescribed convergence rate (that serevs as a performance index, represented by a scalar number).

This performance index is checked locally (with car onboard sensors, e.g., by Mobileye)

If a valid performance index is not measured, local autonomous driving controllers, are activated immediately

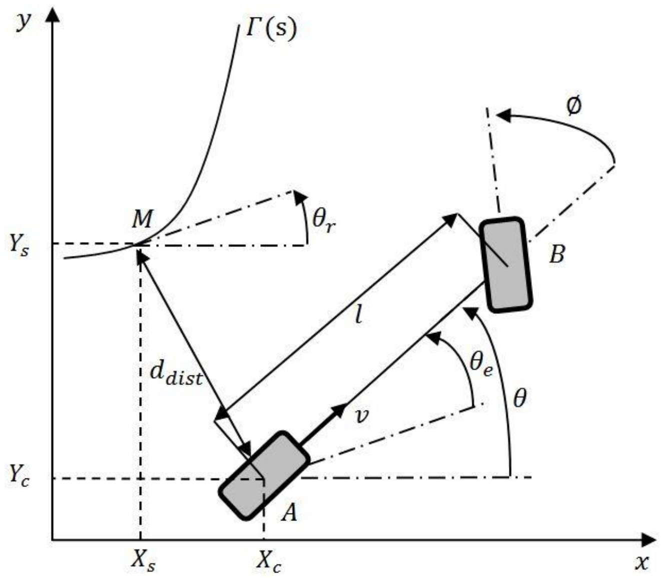


Coordinated path following - Problem formulation

Let $\Gamma(s)$ be a smooth curve (parameterized by s), and a point M denotes the closest point between the curve and the vehicle.

The following model describes the relative state between the vehicle and the path.

$$\begin{bmatrix} \dot{s} \\ \dot{d}_{dist} \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v\dfrac{\cos(\theta_r - \theta)}{1 + d_{dist}c(s)} \\ v\sin(\theta_r - \theta) \\ \dfrac{v\tan(\phi)}{l} - \dfrac{c(s)\cos(\theta_r - \theta)}{1 + d_{dist}c(s)} \end{bmatrix}$$



where $c(s)$ represents the curvature of $\Gamma(s)$

$\theta_e = \theta - \theta_r$ is the relative angle, and d_{dist} is the distance of the vehicle from the path.

Coordinated path following - Problem formulation

The goal is to solve two complementary problems:

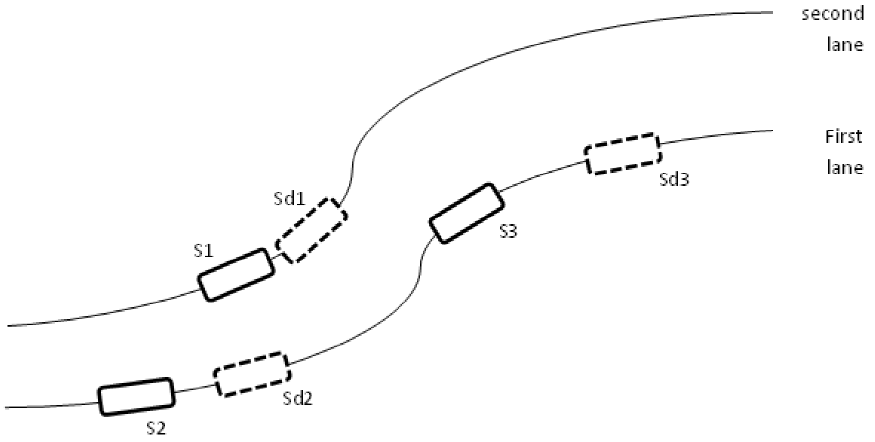
- An inner layer solves the path following control problem of each individual vehicle. This controller ensures that each vehicle converges to the desired path.
- An outer layer solves the coordination control problem for the all group, by setting the velocity of virtual vehicles (denoted by $\dot{s}_d$) which serve as reference vehicles for the real vehicles.

In short, two control signals v and τ are to be designed for each vehicle such that,

$$\lim_{t\to\infty} \|d_{dist}\| = 0$$

$$\lim_{t\to\infty} \|\theta_e\| = 0$$

$$\lim_{t\to\infty} \|s - s_d\| = 0$$



Path following of individual vehicles

The inner layer controller is derived by **the chained form** approach

The tracking errors between the vehicle and the path (at M) are defined as,

distance error $e_1 = d_{dist}$

heading error $e_2 = \sin(\theta_e)$

Steering angle error $e_3 = \dfrac{c(s)\cos^2(\theta_e)}{1+d_{dist}c(s)} - \dfrac{\tan(\phi)\cos(\theta_e)}{l}$

And the steering rate command is,

$$\tau = [\frac{c'(s)\cos^3(\theta_e)}{(1+d_{dist}c(s))^2}v - 3\frac{c^2(s)\cos^2(\theta_e)\sin(\theta_e)}{(1+d_{dist}c(s))^2}v - \frac{\sin(\theta_e)\tan^2(\phi)}{l^2}$$

$$+3v\frac{c(s)\cos(\theta_e)\sin(\theta_e)\tan(\phi)}{l(1+d_{dist}c(s))} - v\frac{d_{dist}c(s)c'(s)\cos^3(\theta_e)}{(1+d_{dist}c(s))^3} - u]\frac{l\cos^2(\phi)}{\cos(\theta_e)}$$

Path following of individual vehicles

These ***complex*** definitions (of tracking errors and control) transforms the system to the ***simple*** chained form model

$$\dot{e}_1 = e_2 v$$
$$\dot{e}_2 = e_3 v$$
$$\dot{e}_3 = u$$

Now, $u(t)$ can be chosen as $u = -Kev = -\left(k_1 e_1 + k_2 e_2 + k_3 e_3\right) v$

Define a weighed square-error function as $V = p_1 e_1^{\ 2} + p_2 e_2^{\ 2} + p_3 e_3^{\ 2}, \quad p_i > 0$

One can choose K such that $\dot{V} \le -\alpha V v$

This is the performance index of the controller.
It is checked locally with car onboard sensors.

Coordinated path following control

On the outer layer, the coordinated problem is managed by defining a virtual vehicle to each controlled vehicle.

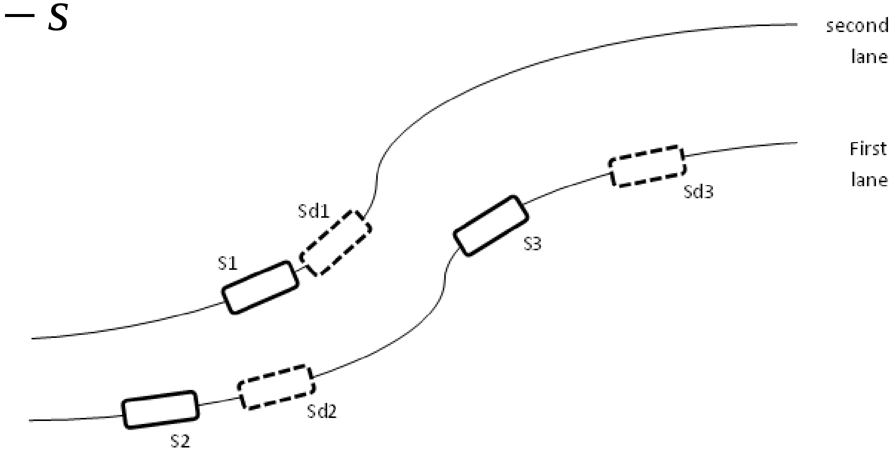
A control law is needed (for each real vehicle) to allow convergence to a virtual vehicle.

The error between a real and virtual vehicle is $s_e = s_d - s$

Taking the linear velocity of the controlled vehicle as,

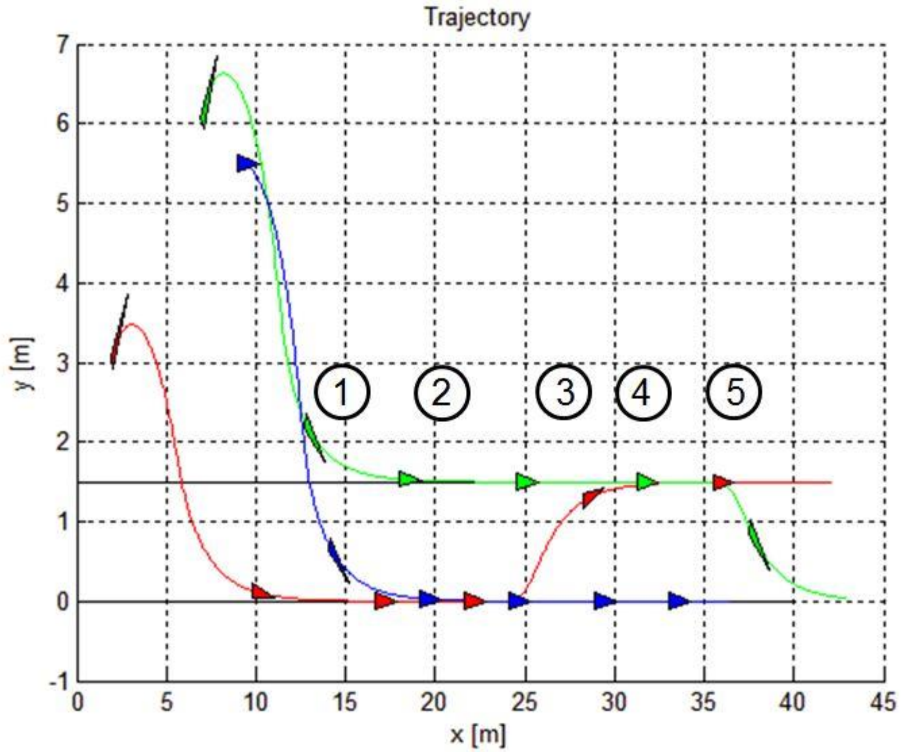
$$v = \frac{1 + d_{dist} c(s)}{\cos(\theta_e)} (\dot{s}_d + \lambda (s_d - s))$$

assures, $\dot{s}_e = -\lambda s_e.$



Simulation results

The simulation illustrates a convergence of three vehicles to a two lane road segment, while driving in different speeds, bypassing each other and avoiding collisions



In the presented case: $N = 3$

fast $[x_1(t_0)\ \ y_1(t_0)\ \ \theta_1(t_0)\ \ \phi_1(t_0)] = [2\ \ 3\ \ \pi/4\ \ 0]^T$

mid $[x_2(t_0)\ \ y_2(t_0)\ \ \theta_2(t_0)\ \ \phi_2(t_0)] = [7\ \ 6\ \ \pi/4\ \ \pi/4]^T$

slow $[x_3(t_0)\ \ y_3(t_0)\ \ \theta_3(t_0)\ \ \phi_3(t_0)] = [9\ \ 5.5\ \ 0\ \ 0]^T$

$[s_0(t_0)\ \ s_{d0}(t_0)]^T = [0\ \ 0]^T$ for all virtual vehicles

Desired speeds are $[v_{d1}\ \ v_{d2}\ \ v_{d3}]^T = [\mathbf{6}\ \ \mathbf{5}\ \ \mathbf{4}]^T$

Vehicle length $l = 1$

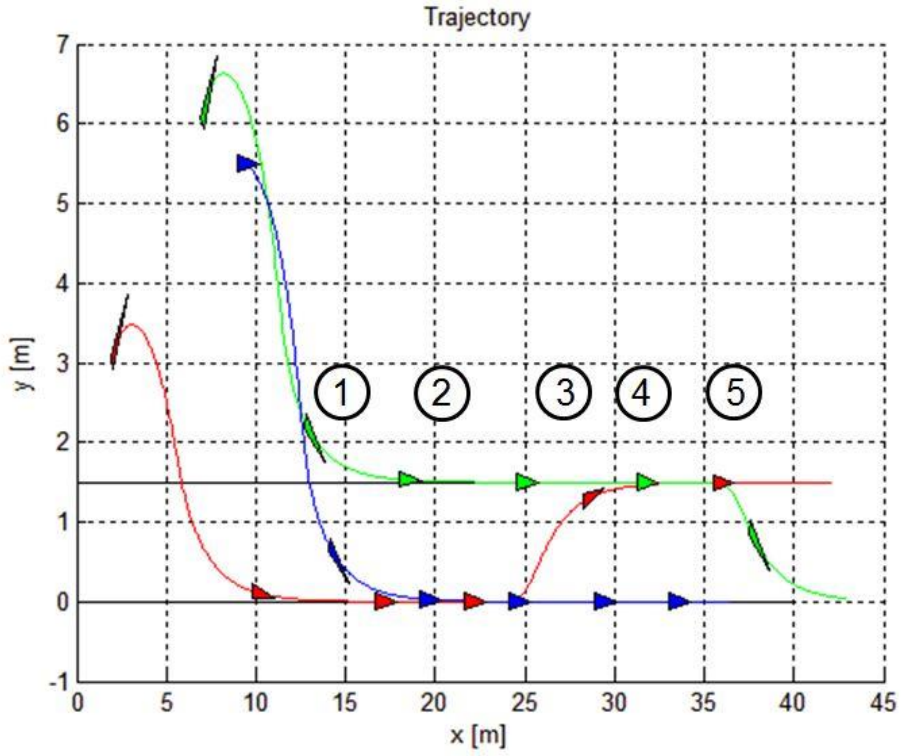
Safety zone $[d\ \ w]^T = [2.5\ \ 0.41]^T$

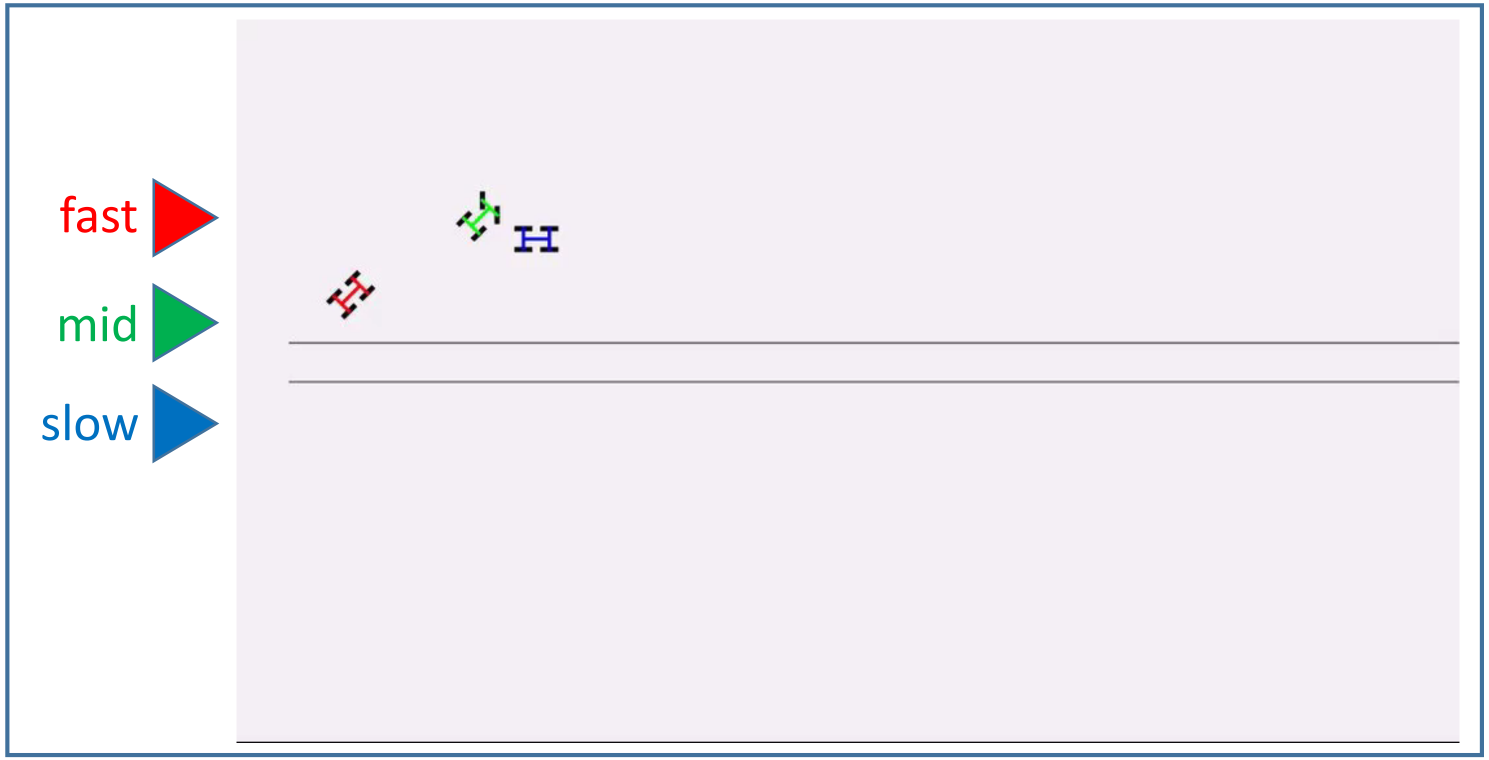
Path following control gains $[k\ \ a_1\ \ a_2]^T = [2\ \ 2\ \ 4]^T$

Coordination control gains $[\lambda\ \ \gamma_1\ \ \gamma_2]^T = [0.2\ \ 0.5\ \ 0.5]^T$

Simulation results

The simulation illustrates a convergence of three vehicles to a two lane road segment, while driving in different speeds, bypassing each other and avoiding collisions





Vehicle length $l = 1$

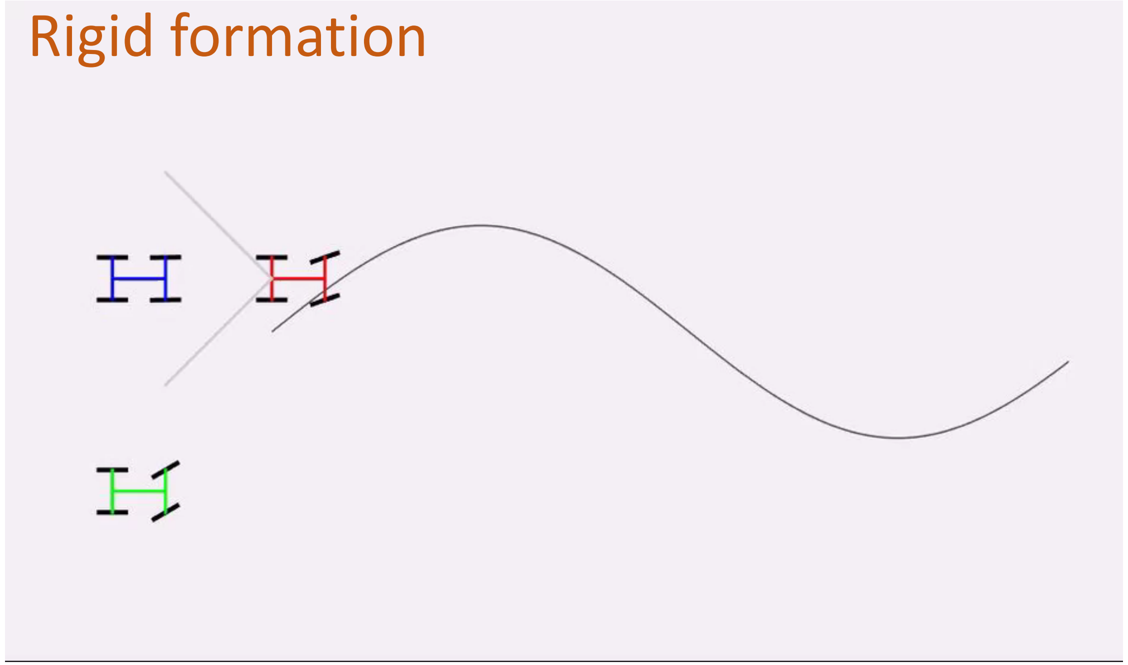
Safety zone $[d \;\; w]^T = [2.5 \;\; 0.41]^T$

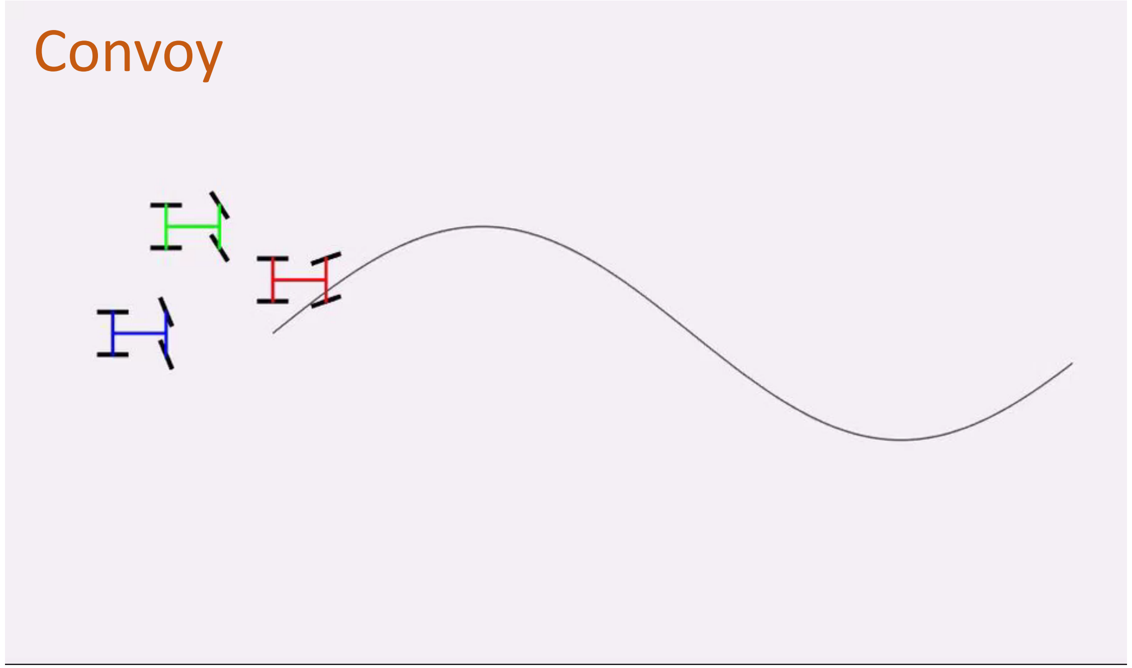
Path following control gains $[k \;\; a_1 \;\; a_2]^T = [2 \;\; 2 \;\; 4]^T$

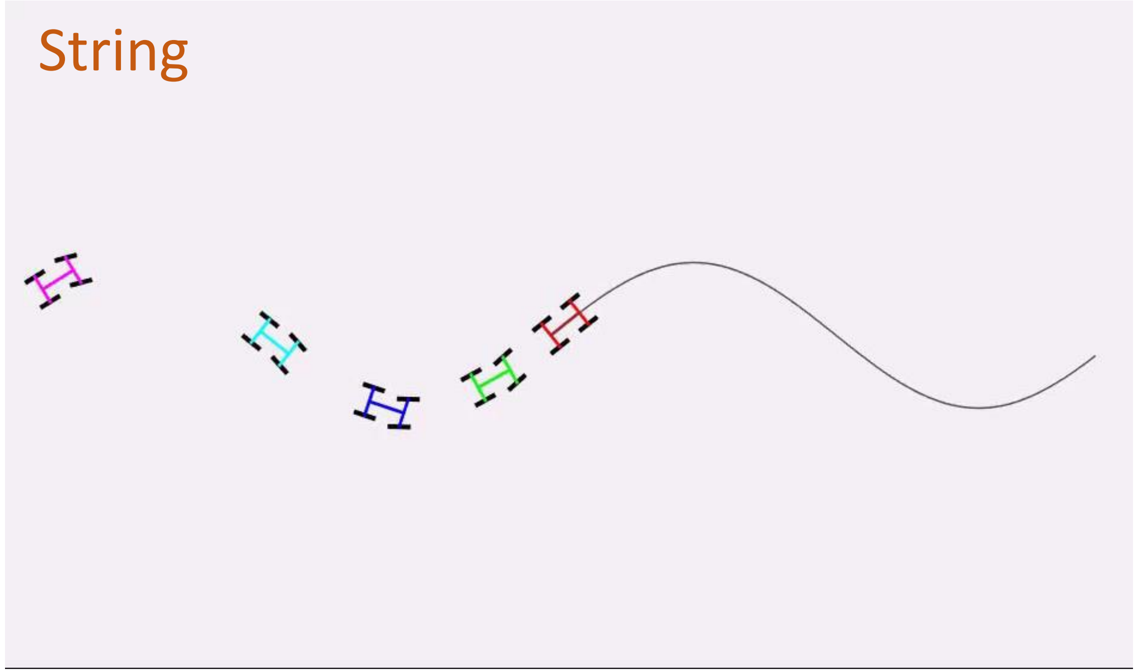
Coordination control gains $[\lambda \;\; \gamma_1 \;\; \gamma_2]^T = [0.2 \;\; 0.5 \;\; 0.5]^T$

Simulation results

Some other coordinated motion examples







Other research activities (but related)

Active magnetic bearings:

- Design (for low losses)
- Control (energy efficiency)
- Applications: FESS, precision motion
- Magnetic levitation

Health Monitoring:

- Detection and isolation of faults
- Model based approach
- Hybrid systems (continuous + discrete)
- Fault tolerant control

Driver assistance systems:

- Active stability (yaw control)
- Active differential
- Electronic differential
- Active anti roll bar

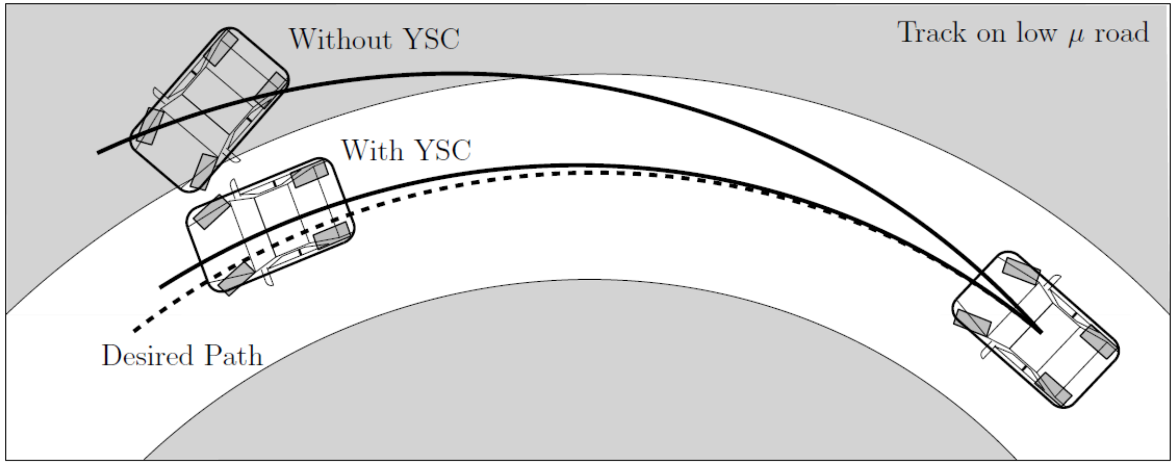
Unmanned Areal Vehicle:

- Flight formation control
- Multi rotor UAVs
- Fault tolerant control
- Unusual designs (flying wing, …)

Yaw Stability Control based on Active differentials

Goal – to prevent vehicles from spinning and drifting-out, due to:

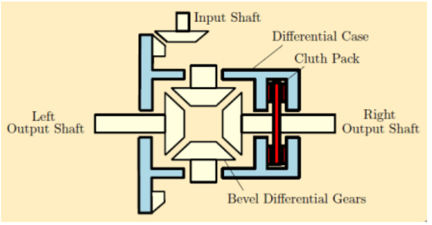
- High acceleration maneuvers.
- Unexpected yaw disturbances.



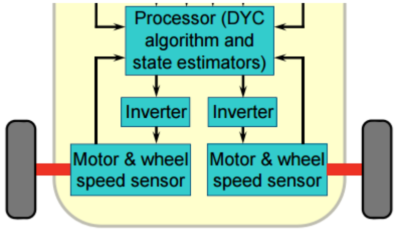
Active differentials - allow electronic control over Left-Right torque transfer:

We develop control algorithm for Two types of active differentials

Active Limited Slip Differential (for a single engine vehicle)



Electronic differential (for Electric Vehicles with Independent Motors



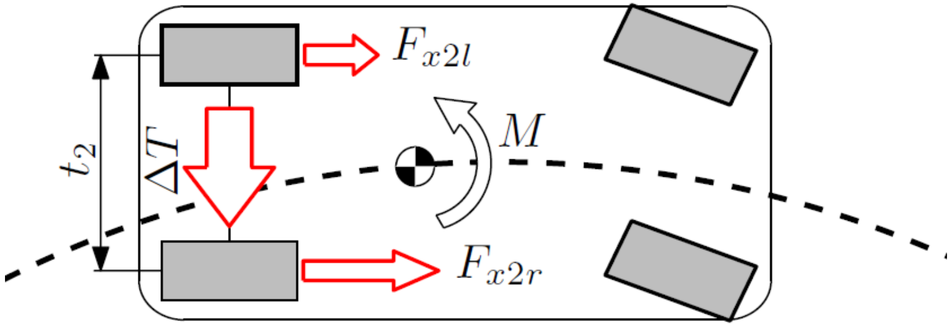
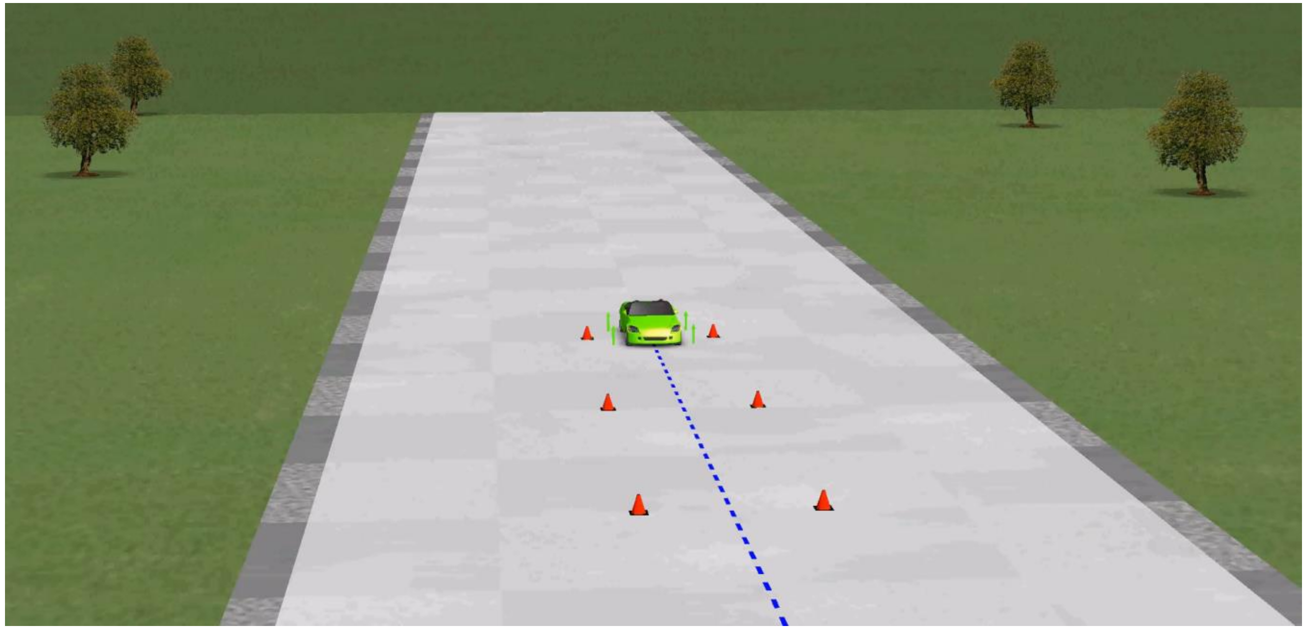
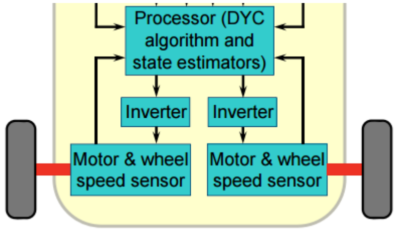


Figure: The function of a rear active differentials.

Yaw Stability Control based on Active differentials



Electronic differential (for Electric Vehicles with Independent Motors



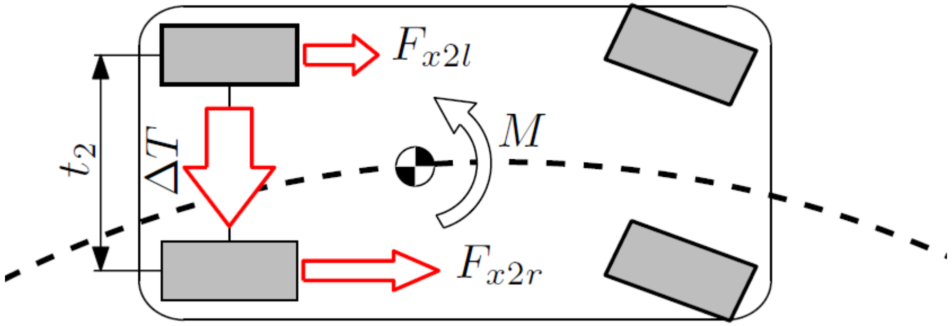


Figure: The function of a rear active differentials.

Health monitoring – Fault detection and Isolation

We are developing model-based algorithms that detect and isolate system faults in real time. The algorithm detects inconsistencies between the measured behavior (by sensors) and the predicted normal behavior (by a model). It also identify the fault type (i.e., fault isolation)

The method has been implemented on the CyCab (electric vehicle) to detect faults of its electro-hydraulic steering system. The considered faults were:

Sensor-faults:
1) Pressure sensors
2) incremental encoder
3) absolute encoder (steering angle).

Sudden faults:
1) A burnt DC-motor or driver
2) A broken-belt

Parametric faults:
1) A flat tire.
2) Piston internal leakage (due to worn-seal, the piston efficiency is decreased)