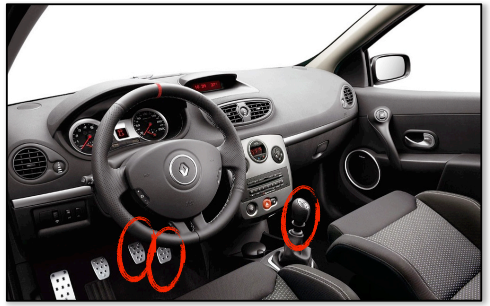
Examples of hybrid MPC

Hybrid MPC for cruise control



GOAL:

command gear ratio, gas pedal, and brakes to **track** a desired speed and minimize consumption

Hybrid model

- Vehicle dynamics

$$\boxed{m\ddot{x} = F_e - F_b - \beta\dot{x}}$$

$\dot{x}$ = vehicle speed

F_e = traction force

F_b = brake force

discretized with sampling time $T_s = 0.5$ s

- Transmission kinematics

$$\boxed{\omega = \frac{R_g(i)}{k_s}\dot{x}}$$

$$\boxed{F_e = \frac{R_g(i)}{k_s}C}$$

ω = engine speed

C = engine torque

i = gear

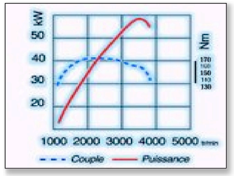
power balance:

$F_e\dot{x} = C\omega$

Hybrid model

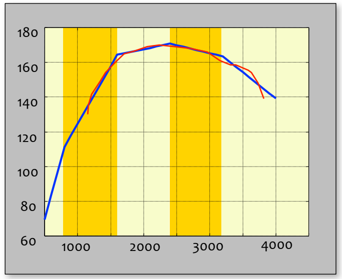
- Engine torque $\boxed{C_e^-(\omega) \leq C \leq C_e^+(\omega)}$

- Max engine torque $C_e^+(\omega)$



Piecewise-linearization

(PWL Toolbox, Julián, 2000)



requires: 4 binary aux variables
4 continuous aux variables

Note: in this case PWA function is concave $\Rightarrow$ could be handled by linear constraints without introducing any binary variable !

- Min engine torque $\boxed{C_e^-(\omega) = \alpha_1\omega + \beta_1}$

Hybrid model

- **Gear selection:** for each gear #i, define a binary input $\boxed{g_i \in \{0,1\}}$
  $i = R, 1, 2, 3, 4, 5$

- **Gear selection (traction force):**

$$\boxed{F_e = \frac{R_g(i)}{k_s} C}$$ depends on gear #i

define auxiliary continuous variables:

$$\text{IF } g_1 = 1 \text{ THEN } F_{ei} = \frac{R_g(i)}{k_s} C \text{ ELSE } 0$$

$$\Rightarrow F_e = F_{eR} + F_{e1} + F_{e2} + F_{e3} + F_{e4} + F_{e5}$$

- **Gear selection (engine/vehicle speed):**

$$\boxed{\omega = \frac{R_g(i)}{k_s} \dot{x}}$$ similarly, also requires 6 auxiliary continuous variables

HYSDEL model

```
SYSTEM cruisecontrolmodel {

INTERFACE {
    PARAMETER {
        REAL mass = 1020; /* kg */
        REAL beta_friction = 25; /* W/m*s */

        [snip]
    }

    STATE { REAL position [0,10000];
            REAL speed [vmin,vmax]; }

    INPUT { REAL torque [Cmin,Cmax];
            REAL F_brake [0,max_brake_force];
            BOOL gear1, gear2, gear3, gear4, gear5, gearR; }
    }

IMPLEMENTATION {
    AUX {REAL F, Fe1, Fe2, Fe3, Fe4, Fe5, FeR;
         REAL w, w1, w2, w3, w4, w5, wR;
         BOOL dPWL1,dPWL2,dPWL3,dPWL4;
         REAL DCe1,DCe2,DCe3,DCe4;  }

    LINEAR {F = Fe1+Fe2+Fe3+Fe4+Fe5+FeR;
            w = w1+w2+w3+w4+w5+wR; }

    AD { dPWL1 = wPWL1-w<=0;
         dPWL2 = wPWL2-w<=0;
         dPWL3 = wPWL3-w<=0;
         dPWL4 = wPWL4-w<=0;  }

    DA { Fe1 = {IF gear1  THEN torque/speed_factor*Rgear1};
         Fe2 = {IF gear2  THEN torque/speed_factor*Rgear2};
         Fe3 = {IF gear3  THEN torque/speed_factor*Rgear3};
         Fe4 = {IF gear4  THEN torque/speed_factor*Rgear4};
         Fe5 = {IF gear5  THEN torque/speed_factor*Rgear5};
         FeR = {IF gearR  THEN torque/speed_factor*RgearR};

         w1 = {IF gear1  THEN speed/speed_factor*Rgear1};
         w2 = {IF gear2  THEN speed/speed_factor*Rgear2};
         w3 = {IF gear3  THEN speed/speed_factor*Rgear3};
         w4 = {IF gear4  THEN speed/speed_factor*Rgear4};
         w5 = {IF gear5  THEN speed/speed_factor*Rgear5};
         wR = {IF gearR  THEN speed/speed_factor*RgearR};

         DCe1 = {IF dPWL1 THEN (aPWL2-aPWL1)+(bPWL2-bPWL1)*w};
         DCe2 = {IF dPWL2 THEN (aPWL3-aPWL2)+(bPWL3-bPWL2)*w};
         DCe3 = {IF dPWL3 THEN (aPWL4-aPWL3)+(bPWL4-bPWL3)*w};
         DCe4 = {IF dPWL4 THEN (aPWL5-aPWL4)+(bPWL5-bPWL4)*w};
    }

    CONTINUOUS { position = position+Ts*speed;
                 speed = speed+Ts/mass*(F-F_brake-beta_friction*speed);

    MUST {  /* max engine speed */
            /* wemin <= w1+w2+w3+w4+w5+wR <= wemax */

        -w1 <= -wemin; w1 <= wemax;
        -w2 <= -wemin; w2 <= wemax;
        -w3 <= -wemin; w3 <= wemax;
        -w4 <= -wemin; w4 <= wemax;
        -w5 <= -wemin; w5 <= wemax;
        -wR <= -wemin; wR <= wemax;

        -F_brake <=0;
        F_brake <= max_brake_force;

        -torque-(alpha1+beta1*w) <=0;
        torque-(aPWL1+bPWL1*w+DCe1+DCe2+DCe3+DCe4)-1<=0;

        -((REAL gear1)+(REAL gear2)+(REAL gear3)+(REAL gear4)+
                          (REAL gear5)+(REAL gearR))<=-0.9999;
        (REAL gear1)+(REAL gear2)+(REAL gear3)+(REAL gear4)+
                          (REAL gear5)+(REAL gearR)<=1.0001;

        dPWL4 -> dPWL3; dPWL4 -> dPWL2;
        dPWL4 -> dPWL1; dPWL3 -> dPWL2;
        dPWL3 -> dPWL1; dPWL2 -> dPWL1;
        }
    }
}
```

go to demo **/demos/cruise/init.m**

Hybrid model

- **MLD model**

$$\begin{aligned} x(t+1) &= Ax(t) + B_1 u(t) + B_2\delta(t) + B_3 z(t) + B_5 \\ y(t) &= Cx(t) + D_1 u(t) + D_2\delta(t) + D_3 z(t) + D_5 \\ E_2\delta(t) + E_3 z(t) &\le E_4 x(t) + E_1 u(t) + E_5 \end{aligned}$$

- 2 continuous states: x, v (vehicle position and speed)
- 2 continuous inputs: C, F_b (engine torque, brake force)
- 6 binary inputs: g_R, g_1, g_2, g_3, g_4, g_5 (gears)
- 1 continuous output: v (vehicle speed)
- 18 auxiliary continuous vars: (6+1 traction force, 6+1 engine speed, 4 PWL max engine torque)
- 4 auxiliary binary vars: (PWL max engine torque breakpoints)
- 100 mixed-integer inequalities

Hybrid controller

- **Max-speed controller**

Objective: maximize speed

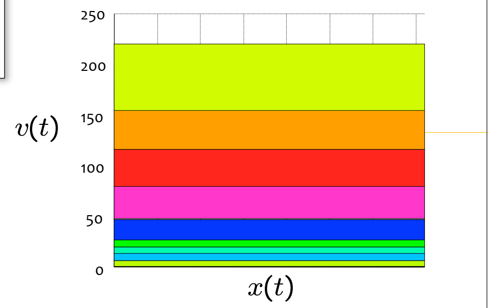
(to reproduce max acceleration plots)

$$\max_{u_t} \; J(u_t, x(t)) \triangleq v(t+1|t)$$

$$\text{s.t.} \quad \begin{cases} \text{MLD model} \\ x(t|t) = x(t) \end{cases}$$

MILP optimization problem

Linear constraints	96
Continuous variables	18
Binary variables	10
Parameters	1
Time to solve mp-MILP (Sun Ultra 10)	45 s
Number of regions	**11**

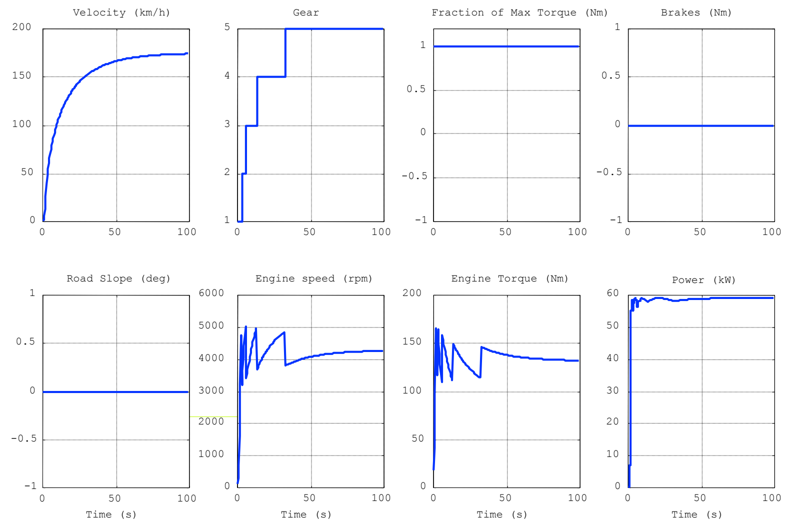


($x(t)$ is irrelevant)

(parameters: Renault Clio 1.9 DTI RXE)

Hybrid controller

- Max-speed controller



Hybrid controller

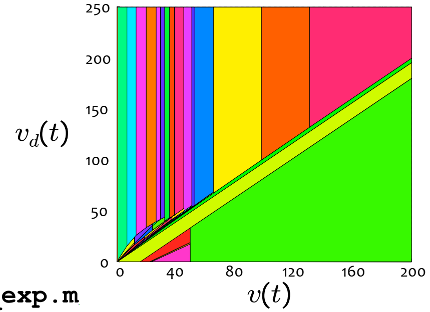
- Tracking controller

$$\min_{u_t} \quad J(u_t, x(t)) \triangleq |v(t+1|t) - v_d(t)| + \rho|\omega|$$

$$\text{s.t.} \quad \begin{cases} \text{MLD model} \\ x(t|t) = x(t) \end{cases}$$

MILP optimization problem

Linear constraints	98
Continuous variables	19
Binary variables	10
Parameters	2
Time to solve mp-MILP (PC 850Mhz)	43 s
Number of regions	**49**

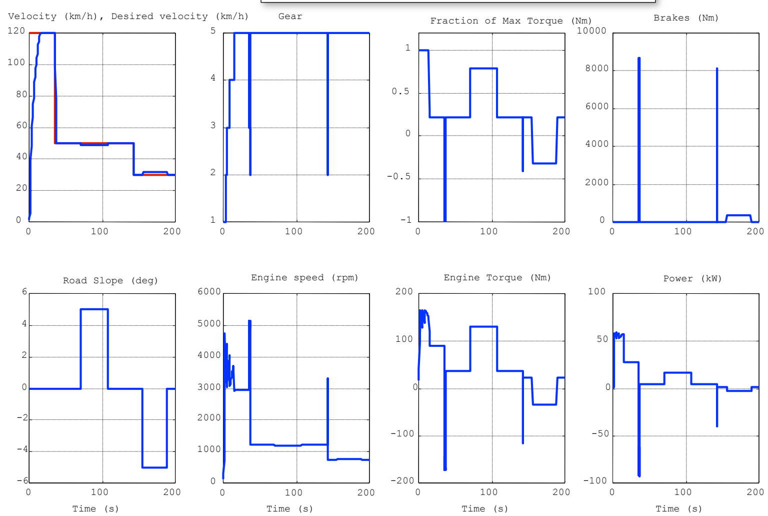


go to demo /**demos/cruise/init_exp.m**

Hybrid controller

- Tracking controller

$$\min_{u_t} |v(t+1|t) - v_d(t)| + \rho|\omega| \qquad \rho = 0.001$$



Hybrid controller

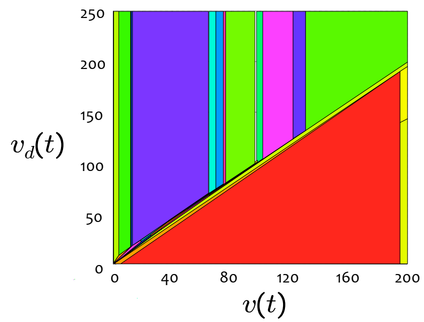
- Smoother tracking controller

$$\min_{u_t} \quad J(u_t, x(t)) \triangleq |v(t+1|t) - v_d(t)| + \rho|\omega|$$

$$\text{s.t.} \quad \begin{cases} \text{MLD model} \\ |v(t+1|t) - v(t)| \leq a_{\max} T_s \\ x(t|t) = x(t) \end{cases}$$

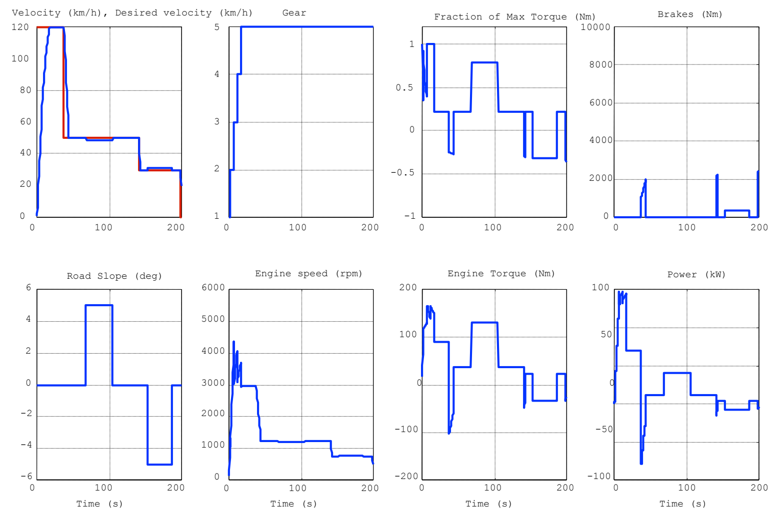
MILP optimization problem

Linear constraints	100
Continuous variables	19
Binary variables	10
Parameters	2
Time to solve mp-MILP (PC 850Mhz)	47 s
Number of regions	**54**



Hybrid controller

- Smoother tracking controller



Traction Control System

Vehicle Traction Control

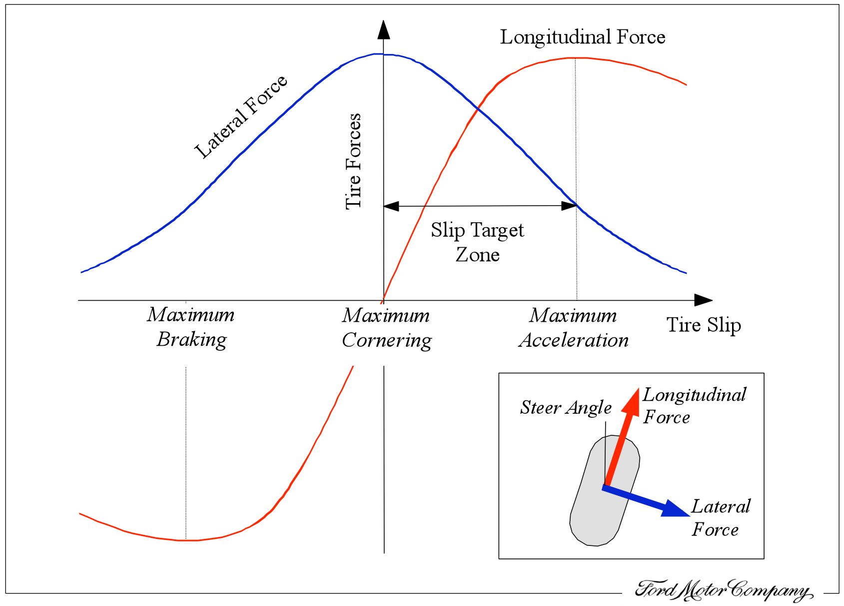
Improve driver's ability to control a vehicle under adverse external conditions (wet or icy roads)

Model: nonlinear, uncertain, constraints

Controller: suitable for real-time implementation

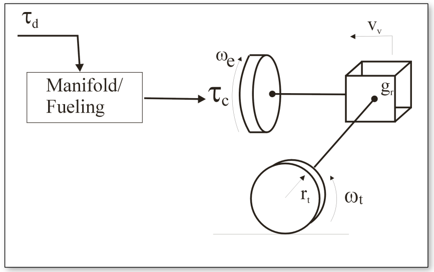
Solution: MLD hybrid framework + explicit hybrid MPC strategy

Tire force characteristics



Simple traction model

(Borrelli, Bemporad, Fodor, Hrovat, 2006)

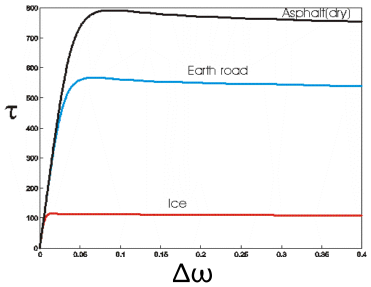


- Mechanical system

$$\dot{\omega}_e = \frac{1}{J_e}\left(\tau_c - b_e\omega_e - \frac{\tau_t}{g_r}\right)$$
$$\dot{v}_v = \frac{\tau_t}{m_v r_t}$$

- Manifold/fueling dynamics

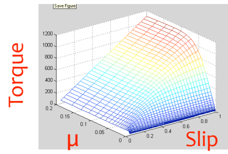
$$\tau_c = b_i \tau_d(t - \tau_f)$$



- Tire torque τ_t is a function of slip Δω and road surface adhesion coefficient μ

$$\Delta\omega = \frac{\omega_e}{g_r} - \frac{v_v}{r_t}$$ wheel slip

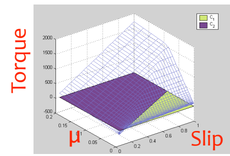
Hybrid model

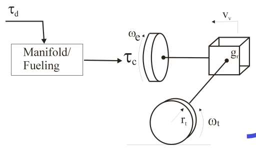


Nonlinear tire torque $\tau_t = f(\Delta\omega, \mu)$

PWA Approximation

(PWL Toolbox, Julian, 2000)





HYSDEL

Mixed-Logical Dynamical (MLD) Hybrid Model (discrete time)

MLD model

$$\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) &\leq E_4x(t) + E_1u(t) + E_5 \end{aligned}$$

state	$x(t) \in \mathbb{R}^4$
input	$u(t) \in \mathbb{R}$
aux. binary	$\delta(t) \in \{0,1\}$
aux. continuous	$z(t) \in \mathbb{R}^3$

number of mixed-integer inequalities = 14

The MLD matrices are automatically generated in MATLAB format by HYSDEL

Performance and constraints

- Control objective:

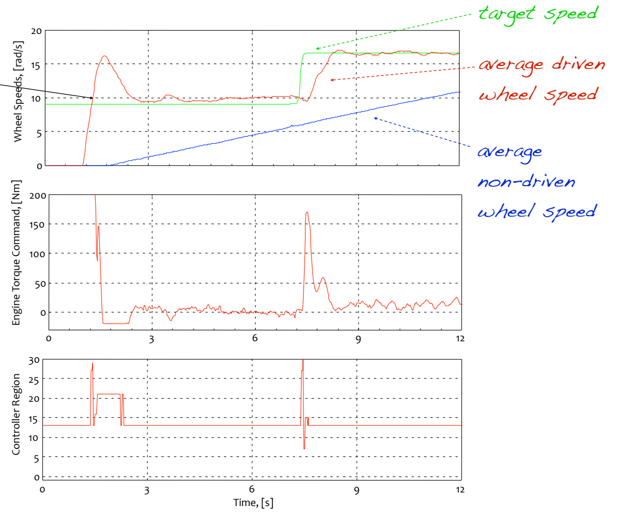
$$\begin{array}{ll} \min & \sum_{k=0}^{N} |\Delta\omega(t+k|t) - \Delta\omega_{\text{des}}| \\ \text{s.t.} & \text{MLD dynamics} \end{array}$$

- Constraints:
  - Limits on the engine torque: $-20\text{ Nm} \leq \tau_d \leq 176\text{ Nm}$

Experimental results

controller is triggered ON

(250 ms delay from commanded to actual engine torque → initial overspin)



Ford Motor Company

Experiments

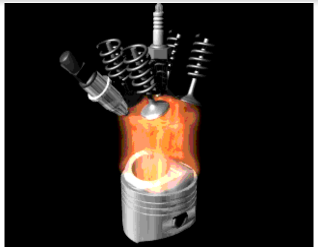
indoor ice arena
(μ≈ 0.2)

2000 Ford Focus
2.0l 4-cyl engine
5-speed manual transmission

- 504 regions
- 20ms sampling time
- Pentium 266Mhz + Labview

Ford Motor Company

Hybrid control of a DISC engine



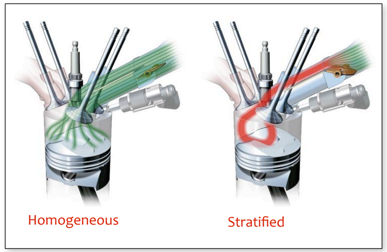
(Photo: Courtesy Mitsubishi)

(N. Giorgetti, G. Ripaccioli, Bemporad, I. Kolmanovsky and D. Hrovat)

DISC engine control problem

Objective: develop a controller for a Direct-Injection Stratified Charge (DISC) engine that:

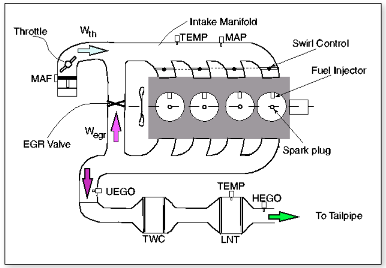
- automatically chooses operating mode (homogeneous/stratified)
- can cope with nonlinear dynamics
- handles constraints on A/F ratio, air-flow, spark
- achieves optimal performance (track desired torque and A/F ratio)



DISC engine

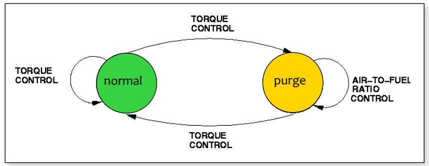
Two distinct regimes:

regime	fuel injection	air-to-fuel ratio
Homogeneous combustion	intake stroke	λ=14.64
Stratified combustion	compression stroke	λ>14.64



- Mode is switched by changing fuel injection timing (late / early)
- Better fuel economy during stratified mode

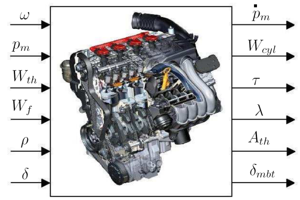
Periodical cleaning of the aftertreatment system needed ($\lambda=14.00$, homogeneous regime)



the stratified operation can only be sustained in a restricted part of the engine operating range

DISC engine

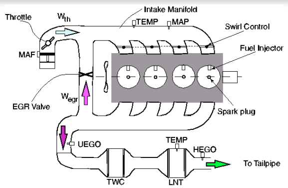
- States: intake manifold pressure (p_m)
- Outputs: Air-to-fuel ratio (λ), torque (τ), max-brake-torque spark timing (δ_{mbt})
- Continuous inputs: spark advance (δ), air flow (W_{th}), fuel flow (W_f)
- Binary input: spark combustion regime (ρ)
- Disturbance: engine speed (ω) [measured]
- Constraints on:
  - Air-to-fuel ratio (due to engine roughness, misfiring, smoke emiss.)
  - Spark timing (to avoid excessive engine roughness)
  - Mass flow rate on intake manifold (constraints on throttle)



Dynamic equations are nonlinear, dynamics and constraints depend on regime ρ

DISC dynamics

Nonlinear model of the engine developed and validated at Ford (Kolmanovsky, Sun, …)



Assumptions:
- no EGR (exhaust gas recirculation) rate,
- engine speed=2000 rpm.

- Intake manifold pressure: $\dot{p}_m = c_m\left(W_{th} - W_{cyl}\right) = c_m\left(W_{th} - k_{cyl}p_m\right)$
- In-cylinder Air-to-Fuel ratio: $\lambda = \dfrac{W_{cyl}}{W_f} = \dfrac{k_{cyl}p_m}{W_f}$
- Engine torque: $\tau = \tau_{mfr} + \tau_{pump} + \tau_{ind}$ with τ_{mfr}, τ_{pump} functions of p_m

$\tau_{ind} = (\theta_a + \theta_b(\delta - \delta_{mbt})^2)W_f$ where θ_a, θ_b, δ_{mbt} are functions of λ, δ and ρ

✓ Good for simulation
✖ Not suitable for optimization-based controller synthesis

Hybridization of DISC model

DYNAMICS (intake pressure, air-to-fuel ratio, torque):

- Definition of two operating points;
- Linearization of nonlinear dynamics;
- Time discretization of the linear models.

ρ-dependent dynamic equations

CONSTRAINTS on:

- Air-to-Fuel Ratio: $\lambda_{min}(\rho) \leq \lambda(t) \leq \lambda_{max}(\rho)$
- Mass of air through the throttle: $0 \leq W_{th} \leq K$
- Spark timing: $0 \leq \delta(t) \leq \delta_{mbt}(\lambda, \rho)$

ρ−dependent constraints

Hybrid system with 2 modes (switching affine system)

Integral action

Integrators on torque error and air-to-fuel ratio error are added to obtain zero offsets in steady-state:

$$\begin{aligned}\epsilon_{\tau,k+1} &= \epsilon_{\tau,k} + T_s(\tau_{\text{ref}}(t) - \tau_k)\\ \epsilon_{\lambda,k+1} &= \epsilon_{\lambda,k} + T_s(\lambda_{\text{ref}}(t) - \lambda_k)\end{aligned}$$

T_s = sampling time

$\tau_{ref},\ \lambda_{ref}$ = references on brake torque and air-to-fuel ratio

Simulation based on nonlinear model confirms zero offsets in steady-state (despite the model mismatch)

MPC of DISC engine

$$\min_{\xi} J(\xi, x(t)) = \sum_{k=0}^{N-1} u_k' R u_k + y_k' Q y_k + x_{k+1}' S x_{k+1}$$

$$\text{subj. to } \begin{cases} x_0 = x(t), \\ \text{hybrid model} \end{cases}$$

N= control horizon

$x(t)$ = current state

$$\xi = [u_0', \gamma_0', z_0', \ldots, u_{N-1}', \gamma_{N-1}', z_{N-1}']',$$

where:

$$\begin{aligned} u_k &= [W_{th,k} - W_{th,ref},\ W_{f,k} - W_{f,ref}, \delta_k - \delta_{ref},\ \rho_k - \rho_{ref}]' \\ y_k &= [\tau_k - \tau_{ref},\ \lambda_k - \lambda_{ref},\ \delta_{mbt} - \delta_k - \Delta\delta_{ref}]' \\ x_k &= [p_{m,k} - p_{m,ref},\ \epsilon_{\tau,k},\ \epsilon_{\lambda,k}]' \end{aligned}$$

and:

$$R = \begin{pmatrix} r_{W_{th}} & 0 & 0 & 0 \\ 0 & r_{W_f} & 0 & 0 \\ 0 & 0 & r_\delta & 0 \\ 0 & 0 & 0 & r_\rho \end{pmatrix} \quad Q = \begin{pmatrix} q_\tau & 0 & 0 \\ 0 & q_\lambda & 0 \\ 0 & 0 & q_{\Delta\delta} \end{pmatrix} \quad S = \begin{pmatrix} s_{p_m} & 0 & 0 \\ 0 & s_{\epsilon_\tau} & 0 \\ 0 & 0 & s_{\epsilon_\lambda} \end{pmatrix}$$

Reference values are automatically generated from τ_{ref} and λ_{ref} by numerical computations based on the nonlinear model

DISC engine - HYSDEL list

```
SYSTEM hysdisc{
  INTERFACE{
  STATE{
    REAL pm      [1, 101.325];
    REAL xtau    [-1e3, 1e3];
    REAL xlam    [-1e3, 1e3];
    REAL taud    [0,    100];
    REAL lamd    [10,    60];
     }
  OUTPUT{
    REAL lambda, tau, ddelta;
         }
  INPUT{
    REAL Wth     [0,38.5218];
    REAL Wf      [0,      2];
    REAL delta   [0,     40];
    BOOL rho;
     }
  PARAMETER{
    REAL Ts, pm1, pm2;
    …
     }
  }

  IMPLEMENTATION{
  AUX{
    REAL lam,taul,dmbtl,lmin,lmax;
     }
  DA{
  lam={IF rho THEN l11*pm+l12*Wth...
             +l13*Wf+l14*delta+l1c
       ELSE   l01*pm+l02*Wth+l03*Wf...
             +l04*delta+l0c      };

    taul={IF rho THEN tau11*pm+...
      tau12*Wth+tau13*Wf+tau14*delta+tau1c
          ELSE   tau01*pm+tau02*Wth...
           +tau03*Wf+tau04*delta+tau0c };


    dmbtl ={IF rho THEN dmbt11*pm+dmbt12*Wth...
            +dmbt13*Wf+dmbt14*delta+dmbt1c+7
            ELSE dmbt01*pm+dmbt02*Wth...
            +dmbt03*Wf+dmbt04*delta+dmbt0c-1};


    lmin ={IF rho THEN 13 ELSE 19};
    lmax ={IF rho THEN 21 ELSE 38};
         }
    CONTINUOUS{
        pm=pm1*pm+pm2*Wth;
        xtau=xtau+Ts*(taud-taul);
        xlam=xlam+Ts*(lamd-lam);
        taud=taud; lamd=lamd;
             }
    OUTPUT{
       lambda=lam-lamd;
       tau=taul-taud;
       ddelta=dmbtl-delta;
       }
    MUST{
       lmin-lam     <=0;
       lam-lmax     <=0;
       delta-dmbtl  <=0;
       }
    }
  }
```


MPC – Torque control mode

$$\min \sum_{k=0}^{N-1} (u_k - u_r)' R (u_k - u_r) + (y_k - y_r)' Q (y_k - y_r) + (x_{k+1} - x_r)' S (x_{k+1} - x_r)$$

$$\text{subj. to } \begin{cases} x_0 = x(t), \\ \text{hybrid model} \end{cases}$$

Solve **MIQP problem** to compute **u(t)**

$$u(t) = [W_{th}(t),\ W_f(t),\ \delta(t),\ \rho(t)]$$

Weights:

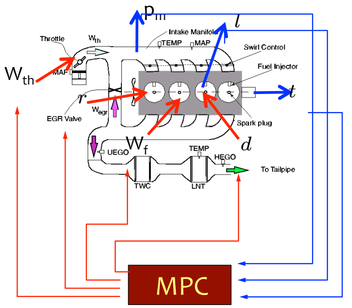
$$R = \begin{pmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

r_ρ (prevents unneeded chattering)

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.01 \end{pmatrix} \quad S = \begin{pmatrix} 0.04 & 0 & 0 \\ 0 & 1500 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

q_τ q_λ s_{ϵ_τ} s_{ϵ_λ}

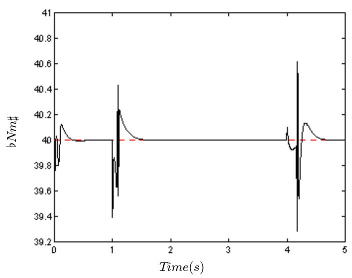
main emphasis on torque



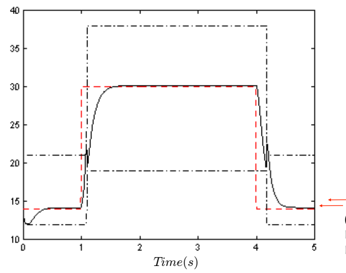
Simulation results (nominal engine speed)

Engine brake torque

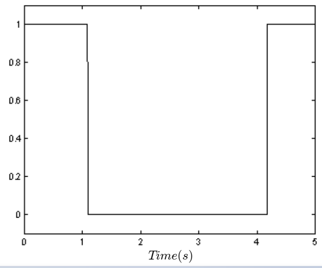
ω/2000 rpm



Air-to-fuel ratio



Combustion mode

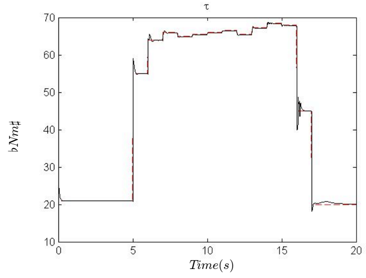


- Control horizon **N=1;**
- Sampling time **Ts=10 ms;**
- PC Xeon 2.8 GHz + Cplex 9.1

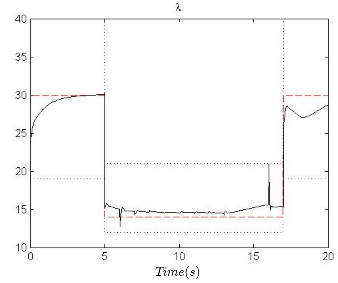
≈ **3 ms** per time step

Simulation results (varying engine speed)

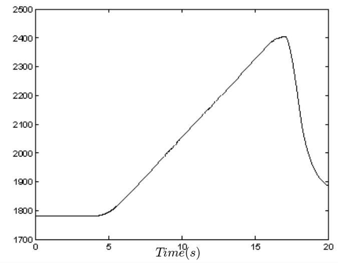
Engine Brake Torque



Air-to-Fuel Ratio



Engine speed

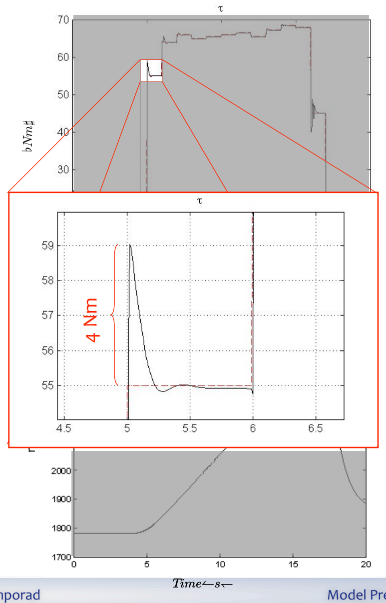


20 s segment of the European drive cycle (NEDC)

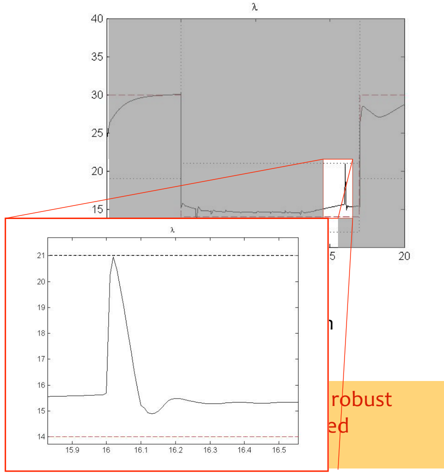
Hybrid MPC design is quite robust with respect to engine speed variations

Simulation results (varying engine speed)

Engine Brake Torque



Air-to-Fuel Ratio



Control code too complex

(MILP) → not implementable !

Explicit MPC controller

Explicit control law: $u(t) = f(\theta(t))$

where: $u = [W_{th}\ W_f\ \delta\ \rho]'$

$$\theta = [p_m\ \epsilon_\tau\ \epsilon_\lambda\ \tau_{\rm ref}\ \lambda_{\rm ref}\ p_{m,\rm ref}\ W_{th,\rm ref}\ W_{f,\rm ref}\ \delta_{\rm ref}]'$$

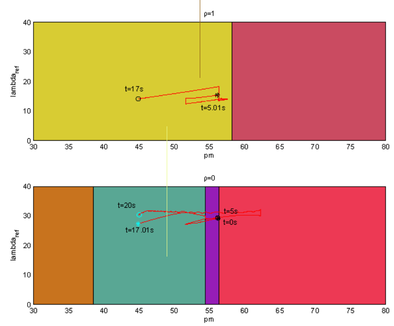
N=1 (control horizon)

42 partitions

- Time to compute explicit MPC: ≈ 3s;
- Sampling time Ts=10 ms;
- PC Xeon 2.8 GHz + Cplex 9.1

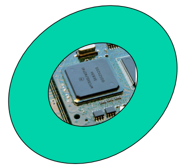
→ **8 μs** per time step

Cross-section by the $\tau_{\rm ref}$-$\lambda_{\rm ref}$ plane



≈3ms on

μ−controller
Motorola
MPC 555
43kb RAM
(custom made for Ford)



Explicit MPC controller (N=2)

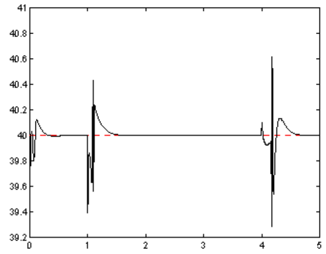
Explicit control law: $u(t) = f(\theta(t))$

where: $u = [W_{th}\ W_f\ \delta\ \rho]'$
$\theta = [p_m\ \epsilon_T\ \epsilon_\lambda\ \tau_{\text{ref}}\ \lambda_{\text{ref}}$
$p_{m,\text{ref}}\ W_{th,\text{ref}}\ W_{f,\text{ref}}\ \delta_{\text{ref}}]'$

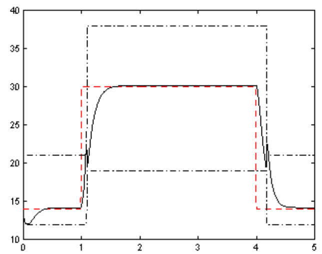
N=2 (control horizon)

747 partitions

Engine Brake Torque



Air-to-Fuel Ratio

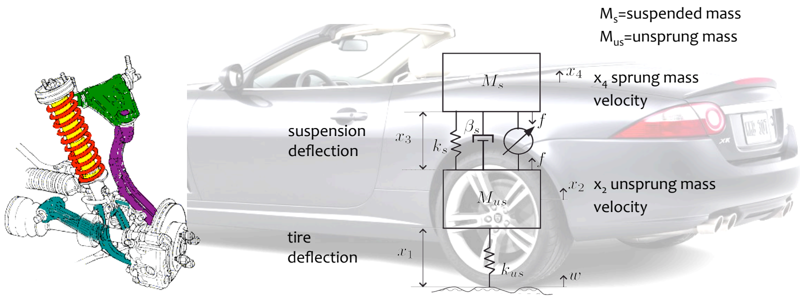


Closed-loop N=2 ⟷ Closed-loop N=1 ⟵ adequate !

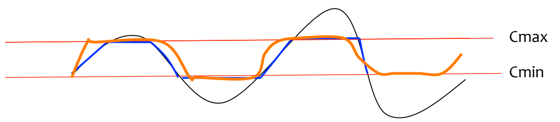
Explicit Hybrid MPC of Semiactive Suspensions

N. Giorgetti, A. Bemporad, H. E. Tseng, and D. Hrovat, "*Hybrid model predictive control application towards optimal semi-active suspension*," International Journal of Control, vol. 79, no. 5, pp. 521–533, 2006.

Quest of optimal semi-active suspensions



For Semi-Active with Variable Damping, $f(x)=C^*(x_4-x_2)$



$C=f(x)/(x_4-x_2)$, where $f(x)$ is the optimal active suspension force

$C=\text{sat}[f(x)/(x_4-x_2)]$

Optimal

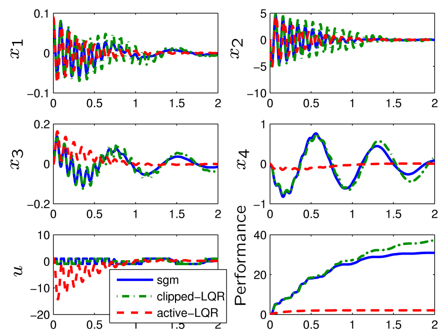
Suboptimal semiactive suspension control

Steepest Gradient (SGM): $\bar{f}_{\text{SGM}} = sat[K_{\text{SGM}}x]$

"Improve the action of a passive suspension"

Shock test of initial condition

$x_0 = [0.09\ 0\ 0\ 0]'$



SGM **16%** better than clipped-LQR

Clipped-LQR is at least 16% from the true optimal

Model

- State-space model $\quad \boxed{\dot{x} = Ax + B\bar{f} + B_w w}$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_{us}^2 & -2\rho\zeta\omega_s & \rho\omega_s^2 & 2\rho\zeta\omega_s \\ 0 & -1 & 0 & 1 \\ 0 & 2\zeta\omega_s & -\omega_s^2 & -2\zeta\omega_s \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \rho \\ 0 \\ -1 \end{bmatrix} \qquad B_w = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

x_1 = tire deflection from equilibrium
x_2 = unsprung mass velocity
x_3 = suspension deflection from equilibrium
x_4 = sprung mass velocity
$\bar{f}$ = normalized adjustable force
w = road velocity disturbance

$$\rho = \frac{M_s}{M_{us}},\ \omega_{us} = \sqrt{\frac{k_{us}}{M_{us}}},\ \omega_s = \sqrt{\frac{k_s}{M_s}},\ \zeta = \frac{\beta_s}{2\sqrt{M_s k_s}},\ \bar{f} = \frac{f}{M_s}$$

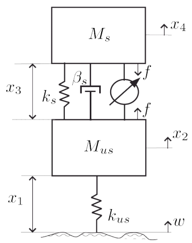
- Output:

$$\boxed{y = \frac{dx_4}{dt} = \begin{bmatrix} 0 & 2\zeta\omega_s & -\omega_s^2 & -2\zeta\omega_s \end{bmatrix} x - \bar{f}}$$

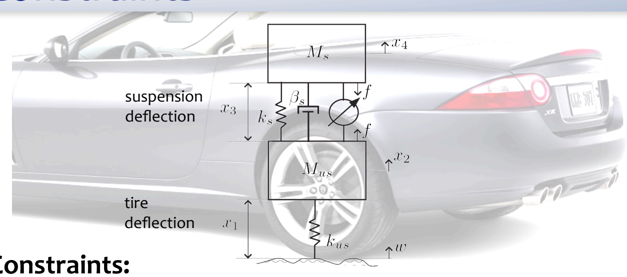
- Cost: $\quad J = \int (q_{x_1} x_1^2 + q_{x_3} x_3^2 + \dot{x}_4^2) dt$

$$= \int (x' Q x + \dot{x}_4^2) dt$$

- Time-discretization: $\quad T_s = 10$ ms



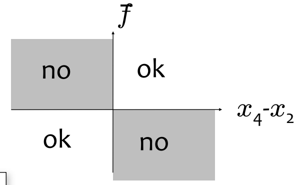
Constraints



Quarter-car model → linear model

Constraints:

1) Passivity condition: $\quad \boxed{\bar{f}(x_4 - x_2) \geq 0}$



2) Max dissipation power:

$$\boxed{\bar{f}(x_4 - x_2) \leq (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2)^2}$$

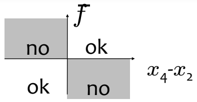
3) Saturation: $\quad \boxed{|\bar{f}| \leq \sigma}$

(1), (2) are nonlinear & nonconvex physical constraints → **hybrid model**

Constraints

1) Passivity condition:

$$\boxed{\bar{f}(x_4 - x_2) \geq 0} \quad \Longleftrightarrow \quad \boxed{\begin{array}{l} [\delta_v = 1] \leftrightarrow [x_4 - x_2 \geq 0] \\ [\delta_{\bar{f}} = 1] \leftrightarrow [\bar{f} \geq 0] \\ [\delta_v = 1] \rightarrow [\delta_{\bar{f}} = 1] \\ [\delta_v = 0] \rightarrow [\delta_{\bar{f}} = 0] \end{array}}$$



2) Max dissipation power:

$$\boxed{\bar{f}(x_4 - x_2) \leq (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2)^2} \quad \Longleftrightarrow \quad \boxed{F \geq 0}$$

where

$$\boxed{F = \begin{cases} \bar{f} - (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2) & \text{if } (x_4 - x_2) \leq 0 \\ -\bar{f} + (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2) & \text{otherwise} \end{cases}}$$

3) Saturation:

$$\boxed{|\bar{f}| \leq \sigma} \quad \Longleftrightarrow \quad \boxed{\begin{array}{lll} \bar{f} & \leq & \sigma \\ \bar{f} & \geq & -\sigma \end{array}}$$

HYSDEL model

```
/* Semiactive suspension system

   (C) 2003-2005 by A.Bemporad, D.Hrovat,
                    E.Tseng, N.Giorgetti
*/

SYSTEM suspension {

 INTERFACE {
    STATE {
        REAL x1 [-0.05,0.05];
        REAL x2 [-5,5];
        REAL x3 [-0.2,0.2];
        REAL x4 [-2,2];
    }
    INPUT{
        REAL u [-10,10];  /* m/s^2 */
    }
    OUTPUT {
        REAL y;
    }
 PARAMETER {
   REAL A1dot,A2dot,A3dot,A4dot,B4dot,ws;
   REAL A11,A12,A13,A14,B1,A21,A22,A23,A24,B2;
   REAL A31,A32,A33,A34,B3,A41,A42,A43,A44,B4;
 }
}
```

```
IMPLEMENTATION {
  AUX {
    BOOL sign;
    BOOL usign;
    REAL F;
  }
  AD {
    sign = x4-x2<=0;
    usign = u<=0;
  }
  DA {
    F={ IF sign THEN u-(2*25.5*ws)*(x4-x2)
                ELSE -u+(2*25.5*ws)*(x4-x2)};
  }
  OUTPUT {    y=A1dot*x1+A2dot*x2+A3dot*x3
               +A4dot*x4+B4dot*u;
  }
  CONTINUOUS {
      x1 = A11*x1+A12*x2+A13*x3+A14*x4+B1*u;
      x2 = A21*x1+A22*x2+A23*x3+A24*x4+B2*u;
      x3 = A31*x1+A32*x2+A33*x3+A34*x4+B3*u;
      x4 = A41*x1+A42*x2+A43*x3+A44*x4+B4*u;
  }
  MUST {
      sign -> usign;
      ~sign -> ~usign;
      F>=0;
  } } }
```

`>>S=mld('semiact3',Ts)` get the MLD model in MATLAB

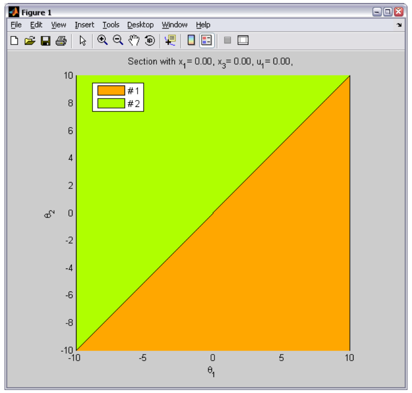
`>>[X,T,D,Z,Y]=sim(S,x0,U);` simulate the MLD model

Hybrid PWA model

- PWA model

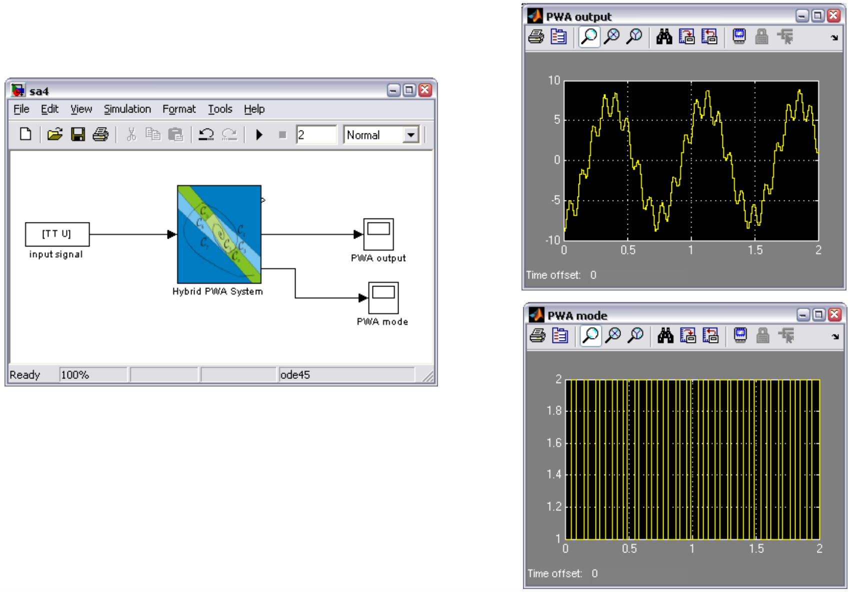
$$\begin{aligned} x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \\ y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \\ i(k) \ \text{s.t.} &\ H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)} \end{aligned}$$

- 4 continuous states
  (x_1, x_2, x_3, x_4)
- 1 continuous input
  (normalized adjustable damping force $\bar{f}$)
- 2 polyhedral regions




```
>>P=pwa(S);
```


Simulation in Simulink



Performance specs

tire deflession

suspension deflession

vertical acceleration

$$\min \left(\sum_{k=1}^{N-1} 1100 x_{1,k}^2 + 100 x_{3,k}^2 + \dot{x}_{4,k}^2 \right) + x_N' P x_N$$

terminal weight (Riccati matrix)

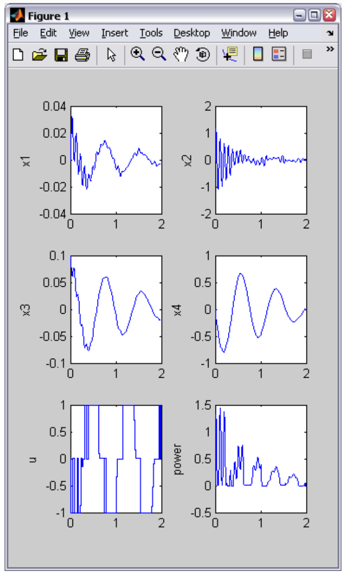
Closed-loop MPC results (command line)

$$J = \int (q_1 x_1^2 + q_3 x_3^2 + \dot{x}_4^2)$$

```
>>refs.y=1;   % weights output #1
>>Q.y=Ts*rx4d;% output weight
…
>>Q.norm=2;   % quadratic costs
>>N=1;        % optimization horizon
>>limits.umin=umin;
>>limits.umax=umax;
```

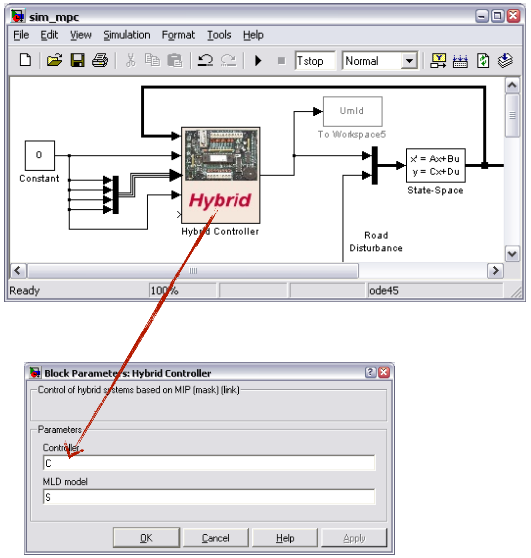
```
>>C=hybcon(S,Q,N,limits,refs);
```

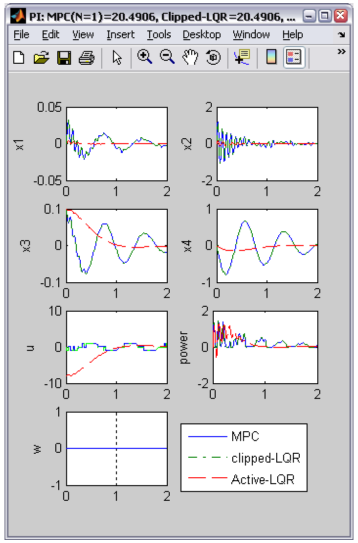
```
>> C

Hybrid controller based on MLD model S <semiact3.hys> [2-norm]

  4 state measurement(s)
  1 output reference(s)
  1 input reference(s)
  4 state reference(s)
  0 reference(s) on auxiliary continuous z-variables

  4 optimization variable(s) (2 continuous, 2 binary)
 13 mixed-integer linear inequalities
sampling time = 0.01, MIQP solver = 'cplex'

Type "struct(C)" for more details.
>>
```




```
>>[XX,UU,DD,ZZ,TT]=sim(C,S,r,x0,Tstop);
```

Closed-loop MPC results (Simulink)

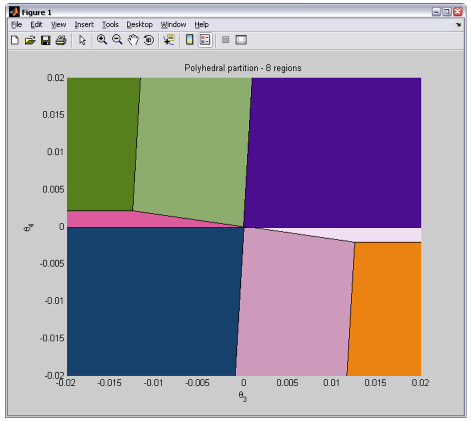




Explicit hybrid MPC

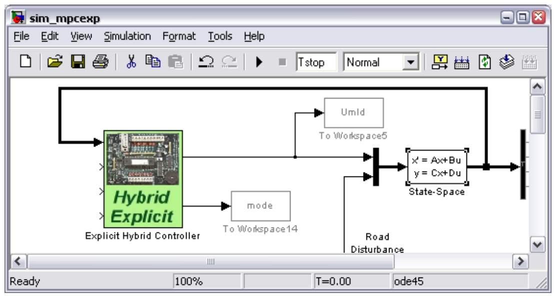
```
>>E=expcon(C,range,options);
```

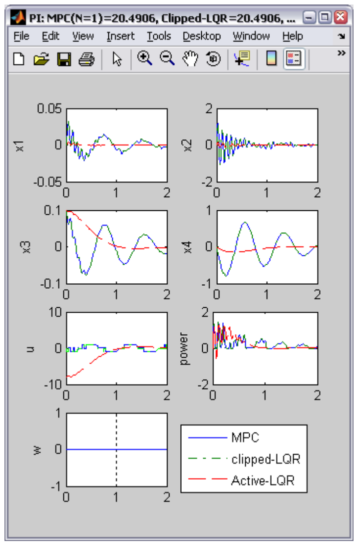
```
>> E

Explicit controller (based on hybrid controller C)
  4 parameter(s)
  1 input(s)
  8 partition(s)
sampling time = 0.01

The controller is for hybrid systems (tracking)
[2-norm]

This is a state-feedback controller.

Type "struct(E)" for more details.
>>
```



Section in the (x_3,x_4)-space for $x_1=x_2=0$

Explicit hybrid MPC

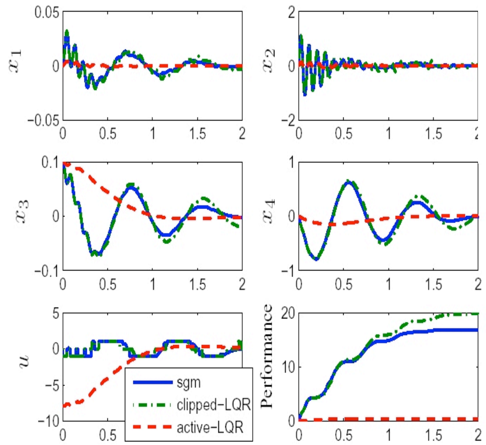




generated C-code

```
#define EXPCON_NU 1
#define EXPCON_NX 4
#define EXPCON_NY 1
#define EXPCON_TS 0.01000000
#define EXPCON_REG 8
#define EXPCON_NTH 4
#define EXPCON_NYM 4
#define EXPCON_NUC 1
#define EXPCON_NUB 0
#define EXPCON_NGAIN 1
#define EXPCON_NH 21
#define EXPCON_NF 8
static double EXPCON_F[]={
    10.4748,0,0,0,10.4748,0,0,0,-0.244594,0,-
    480.664,0,
    3.92349,0,
    480.664,0
    };

static double EXPCON_G[]={
    0,1e-006,-1e-006,-1,0,0,1e-006,1
```


Quest of optimal semiactive suspensions



PARAMETER VALUES USED IN SIMULATION

Parameter	Value	Description
T_s	10 ms	Sampling time
ω_s	1.5 Hz	Sprung mass natural frequency
ω_{us}	10 Hz	Wheel-hop natural frequency
ρ	10	Sprung-to-unsprung mass ratio
ζ	0	Damping ratio
σ	1	Maximum force capacity
q_1	1100	Weight on tire deflection
q_3	100	Weight on suspension deflection

TABLE II

SHOCK TEST: MPC COST VALUE FOR DIFFERENT CONTROL HORIZONS SUBJECTED TO I.C.=[0 0 0.1 0]'.

N	MPC	Clipped-LQR	SGM	LQR
1	20.4282	20.4282	17.4944	0.4446
2	20.4054			
3	20.3290			
4	20.1100			
5	19.7380			
10	20.9840			
12	19.3084			
14	18.4842			
15	18.5996			
16	19.3212			
20	18.0764			
30	17.1494			
40	17.1304			

Simulation results

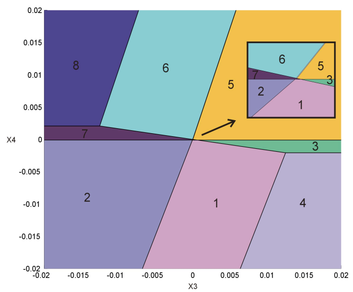
- Horizon N=1: same as Clipped-LQR !
- Better closed-loop performance for increasing N

Performance Index

N	MPC	Clipped-LQR
1	1.5155	1.5155
5	1.4416	
10	1.5238	
15	1.3083	
20	1.2204	
30	1.1456	
40	1.1462	

N=1, same cost value !

Explicit solution (N=1, x_1=x_2=0):



$$u(x) = \begin{cases} 10.4748x_1 + 0.2446x_2 + 79.1519x_3 - 3.9235x_4 \ (= K_{LQ}) & \text{Regions } \#1,\ \#6 \\ 0 & \text{Regions } \#2,\ \#5 \\ (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2) & \text{Regions } \#3,\ \#7 \\ -1 & \text{Region } \#4 \\ 1 & \text{Region } \#8 \end{cases}$$

- Simulations with road noise.
- Initial condition x(0)=[0 0 0 0]'
- Simulation time T=20 s, sampling time T_s=10 ms

Vehicle Yaw Stability Control by Coordinated Active Front Steering and Differential Braking

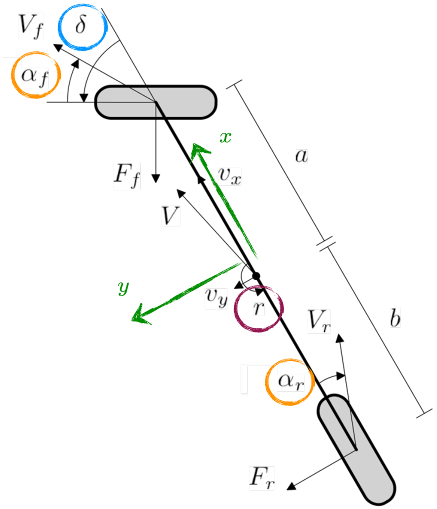
(D. Bernardini, S. Di Cairano, A. Bemporad and H.E. Tseng, CDC 2009)
(S. Di Cairano, H.E. Tseng, D. Bernardini, A. Bemporad, IEEE TCST, 2012)

Overview

- **Problem:**
  Control vehicle stability while tracking driver's desired trajectory
  - **Electronic Stability Control** (ESC)
    (Koibuchi *et al.*, 1996)
  - **Active Front Steering** (AFS)
    (Ackermann, 1997)
- **Main control objective:**
  Force the vehicle **yaw rate** to track a time-varying reference computed by the driver's steering angle and the current vehicle velocity
- **Approach:**
  Consider the steer as a reference generator and actuate steering and differential braking **(coordinated AFS and ESC action)**

Vehicle model

- **Bicycle model** appropriate in high speed turns (Gillespie, 1992)

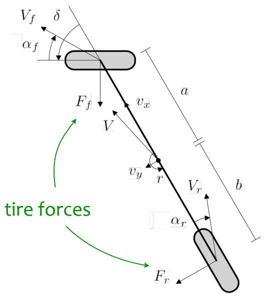


- **reference frame** (x,y,z) moving with the vehicle
- **front steering angle** δ [rad]
- **tire slip angles** α_f,α_r [rad]
- **yaw rate** r [rad/s]

$$\begin{aligned} \tan(\alpha_f + \delta) &= \frac{v_y + ar}{v_x} \\ \tan \alpha_r &= \frac{v_y - br}{v_x} \end{aligned}$$

Tire force model

- **Tire force characteristics:** are nonlinear functions of the **slip angles** and of the **longitudinal slip**



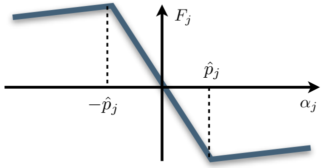
- For a constant longitudinal slip, we use a **piecewise affine model**

$$F_f(\alpha_f) = \begin{cases} -c_f\alpha_f & \text{if } -\hat{p}_f \le \alpha_f \le \hat{p}_f \\ -(d_f\alpha_f + e_f) & \text{if } \quad \alpha_f > \hat{p}_f \end{cases}$$

$$F_r(\alpha_r) = \begin{cases} -c_r\alpha_r & \text{if } -\hat{p}_r \le \alpha_r \le \hat{p}_r \\ -(d_r\alpha_r + e_r) & \text{if } \quad \alpha_r > \hat{p}_r \end{cases}$$

- **Critical slip angles** $\hat{p}_f$, $\hat{p}_r$ are threshold values where dynamics switch

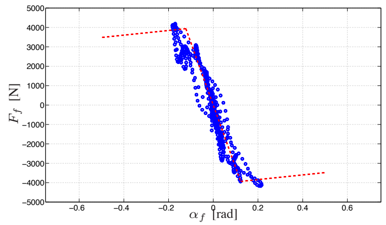
- For symmetry, we can restrict to analyze **clockwise turns** (counter-clockwise turns can be handled by opportunely inverting signs)



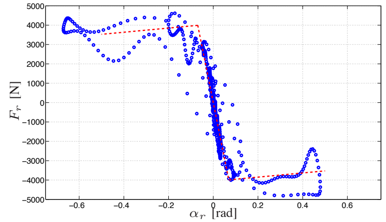
tire forces characteristics

Tire force model

Sideslip angle-force characteristics



(a) Front tires



(b) Rear tires

Experimental tire data and piecewise linear approximation of the tire

Rear-wheel drive test vehicle equipped with active front steering and differential braking used for experimental validation

Dynamical model

slip angles

$$\begin{aligned} \dot{\alpha}_f &= \frac{\dot{v}_y + a\dot{r}}{v_x} - \dot{\delta} \\ \dot{\alpha}_r &= \frac{\dot{v}_y - b\dot{r}}{v_x} \end{aligned}$$

static yaw rate

$$r = \frac{v_x}{a+b}(\alpha_f - \alpha_r + \delta)$$

overall dynamical model

$$\begin{aligned} \dot{\alpha}_f &= \frac{F_f + F_r}{m v_x} - \frac{v_x}{a+b}(\alpha_f - \alpha_r + \delta) + \frac{a}{v_x I_z}(aF_f - bF_r + Y) \\ \dot{\alpha}_r &= \frac{F_f + F_r}{m v_x} - \frac{v_x}{a+b}(\alpha_f - \alpha_r + \delta) - \frac{b}{v_x I_z}(aF_f - bF_r + Y) \\ r &= \frac{v_x}{a+b}(\alpha_f - \alpha_r + \delta) \end{aligned}$$

$$\dot{v}_y = \frac{F_f \cos\delta + F_r}{m} - r v_x$$

lateral velocity

$$\dot{r} = \frac{aF_f \cos\delta - bF_r + Y}{I_z}$$

yaw rate derivative

Piecewise-affine model

- **The overall dynamics model** is recast as a **PWA system** by introducing the Boolean variables

$$\begin{aligned} \gamma_f = 0 \quad &\leftrightarrow \quad \alpha_f \le \hat{p}_f \\ \gamma_r = 0 \quad &\leftrightarrow \quad \alpha_r \le \hat{p}_r \end{aligned}$$

- By **discretizing** with sampling period T_s= 0.1 s we obtain

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C x(k) + D u(k) \\ i \in \{1, \dots, 4\} &: \quad H_i x(k) \le K_i \end{aligned}$$

where $x = [\alpha_f \ \alpha_r]'$ $\quad u = [Y \ \delta]'$ $\quad y = r$

slip angles; yaw moment; front steering; yaw rate

Reference generation

- **Control goal:**
  stabilize the system at the equilibrium obtained with $\delta(k) = \hat{\delta}(k)$
  while **minimizing** the use of the brake actuator $(\hat{Y}(k) = 0)$

  driver's steering angle

- **Equilibrium** condition in the linear region:

$$\begin{aligned}\cancelto{0}{\dot{\alpha}_f} &= \frac{F_f+F_r}{mv_x} - \frac{v_x}{a+b}(\alpha_f - \alpha_r + \cancelto{\hat{\delta}}{\delta}) + \frac{a}{v_x I_z}(aF_f - bF_r + \cancelto{0}{Y})\\ \cancel{\dot{\alpha}_r} &= \frac{F_f+F_r}{mv_x} - \frac{v_x}{a+b}(\alpha_f - \alpha_r + \cancel{\delta}) - \frac{b}{v_x I_z}(aF_f - bF_r + \cancel{Y})\end{aligned}$$

- **Time-varying set-points** are defined using the overall dynamical model

$$\begin{aligned}\hat{\alpha}_f &= \frac{m\tilde{v}_x^2 bc_r\hat{\delta}}{m\tilde{v}_x^2(ac_f - bc_r) - c_f c_r (a+b)^2}\\ \hat{\alpha}_r &= \hat{\alpha}_f \frac{ac_f}{bc_r}\\ \hat{r} &= \frac{\tilde{v}_x}{a+b}(\hat{\alpha}_f - \hat{\alpha}_r + \hat{\delta})\end{aligned}$$

- Current longitudinal velocity $v_x(k)$ is used to **update set-points**

Hybrid prediction model

- **Yaw rate tracking:**
  zero tracking error in steady state is provided by **integral action**

integral of tracking error

$$\begin{aligned}I_r(k+1) &= I_r(k) + r(k) - r_s(k)\\ r_s(k+1) &= r_s(k)\end{aligned}$$

yaw rate set-point

- The **global hybrid dynamical model** of the vehicle is given by

$$\begin{aligned}z(k+1) &= \tilde{A}_i z(k) + \tilde{B}_i u(k) + \tilde{f}_i\\ y(k) &= \tilde{C}z(k) + \tilde{D}u(k)\\ i &\in \{1,\dots,4\} : \tilde{H}_i z(k) \le \tilde{K}_i\end{aligned}$$

augmented state

$$z = \begin{bmatrix}\alpha_f\\ \alpha_r\\ I_r\\ r_s\end{bmatrix}$$

$$\tilde{A}_i = \begin{bmatrix} \multicolumn{2}{c}{A_i} & 0 & 0\\ & & 0 & 0\\ \frac{v_x}{a+b} & \frac{-v_x}{a+b} & 1 & -1\\ 0 & 0 & 0 & 1\end{bmatrix},\ \tilde{B}_i = \begin{bmatrix} \multicolumn{2}{c}{B_i}\\ 0 & \frac{v_x}{a+b}\\ 0 & 0\end{bmatrix},\ \tilde{f}_i = \begin{bmatrix} f_i\\ 0\\ 0\end{bmatrix},$$

$$\tilde{C} = \begin{bmatrix}\frac{v_x}{a+b} & \frac{-v_x}{a+b} & 0 & 0\end{bmatrix},\ \tilde{D} = \begin{bmatrix}0 & \frac{v_x}{a+b}\end{bmatrix},$$

$$\tilde{H}_i = \begin{bmatrix}H_i & 0 & 0\end{bmatrix},\ \tilde{K}_i = K_i\ .$$

Control problem formulation

- The **optimal control problem** solved at every time step k is

$$\begin{aligned}\min_{\mathbf{u}_k}\ & \sum_{j=0}^{N-1} \{(z_{k+j|k} - \hat{z})' Q_z (z_{k+j|k} - \hat{z})\\ & +(y_{k+j|k} - \hat{y})' Q_y (y_{k+j|k} - \hat{y})\\ & +(u_{k+j|k} - \hat{u})' Q_u (u_{k+j|k} - \hat{u})\}\\ \text{s.t.}\ & z_{k|k} = z(k)\\ & \text{hybrid dynamics}\\ & \text{state and input constraints}\end{aligned}$$

slip angles tracking & int. action

yaw rate tracking

penalty on actuators action

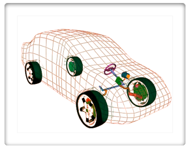
MIQP

- State and input **constraints**:

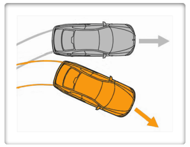
$$\begin{aligned}[z]_1(k) &\ge -\hat{p}_f \quad \text{front slip angle} & |[u]_1(k)| &\le 1000\ [\text{Nm}] \quad \text{yaw moment}\\ [z]_2(k) &\ge -\hat{p}_r \quad \text{rear slip angle} & |[u]_2(k)| &\le 0.35\ [\text{rad}] \quad \text{front steering}\end{aligned}$$

Simulations results

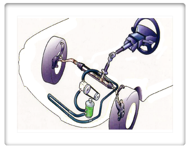
- Simulations run on a **nonlinear vehicle model** including:



longitudinal and lateral vehicle dynamics



yaw rate dynamics



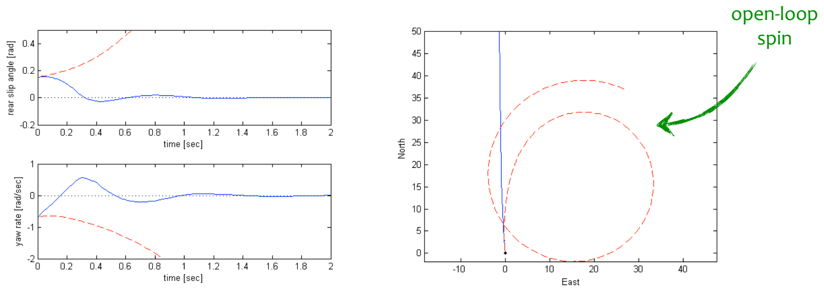
steering actuation dynamics

- Driver's steering command **constant** over the simulation interval
- **Controller setup** after calibration:
  - Prediction horizon $N = 3$
  - Weight matrices $Q_z = \begin{bmatrix} .1 & 0 & 0 & 0\\ 0 & .1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0\end{bmatrix},\quad Q_u = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\quad Q_y = 1$

Simulations results

- **Stability analysis** under nominal conditions (with $\hat{\delta} = 0$):

open-loop vs closed-loop

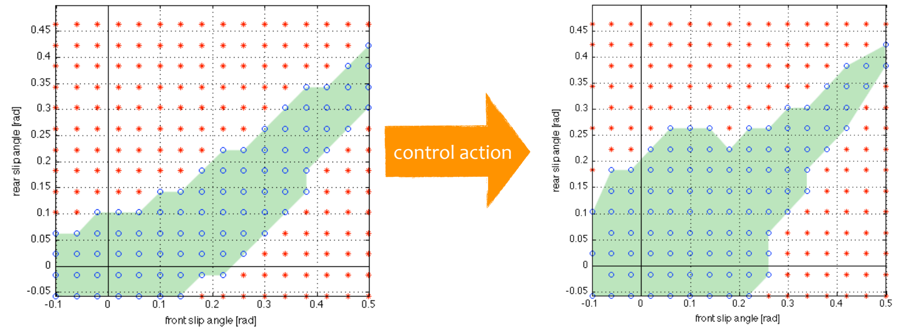


- Controller has to cope with **linearization errors**

Simulations results

- **Stability analysis** under nominal conditions (with $\hat{\delta} = 0$):

open-loop closed-loop



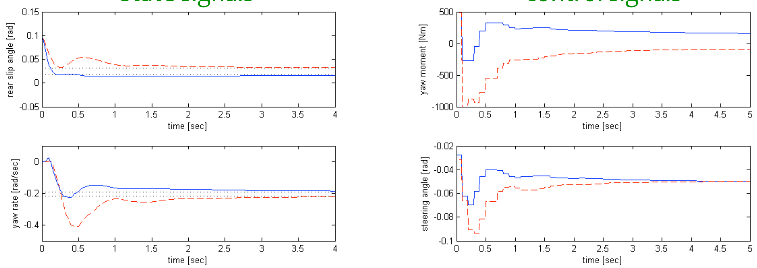
stable initial conditions
unstable initial conditions

closed-loop provides a **larger** stability region

Simulations results

- **Robustness analysis** w.r.t. model mismatches (with $\hat{\delta} = -0.05$):
  - nominal longitudinal velocity $\hat{v}_x = 20 \text{ m/s}$
  - real longitudinal velocity $v_x = 15 \text{ m/s}$ and $v_x = 25 \text{ m/s}$

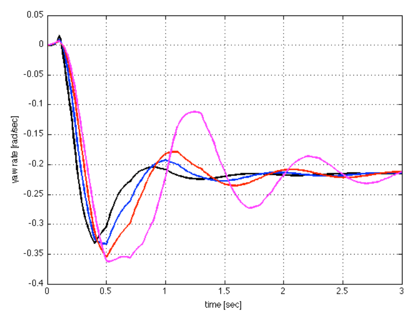
state signals control signals



- Stability and **fast tracking response** are provided

Simulations results

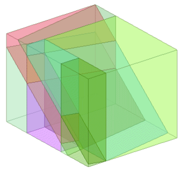
- **Turns on slippery road surface** (with $\hat{\delta} = -0.05$):
  - several values tested: $\hat{s} = 0$, $s = 0.20$, $s = 0.30$, $s = 0.35$



- Good **degree of robustness** with respect to slip mismatches

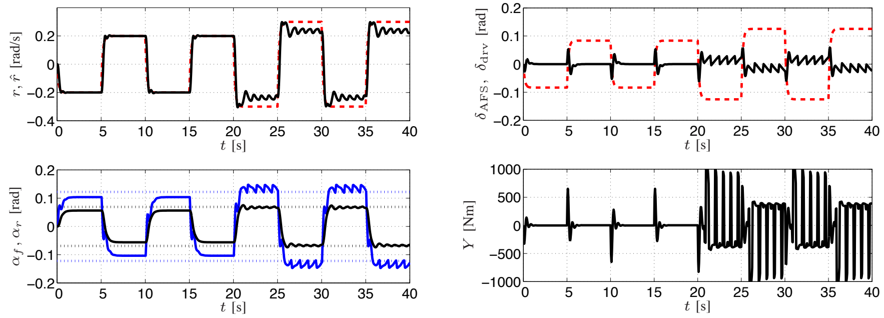
Simulations results

- **Computational issues:**
  - MPC-based approach is **viable for experimental tests** (average CPU time 17ms, worst-case CPU time 63 ms in MATLAB), but requires a MIQP **solver** in the ECU
- **Explicit solution** of hybrid MPC control problem:
  - Exploits **multiparametric programming** techniques to provide a description of the control law as an **explicit** function of the state
  - All the computation is executed **off-line**, only simple set-membership tests and function evaluations are performed on-line to compute the control action
  - However, the explicit solution requires **memory** (around **5000 polytopes** to be stored, in this case!)
  - An **approximation** of the solution is needed for real implementation



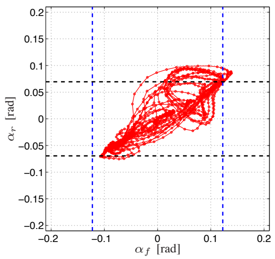
Switched MPC solution

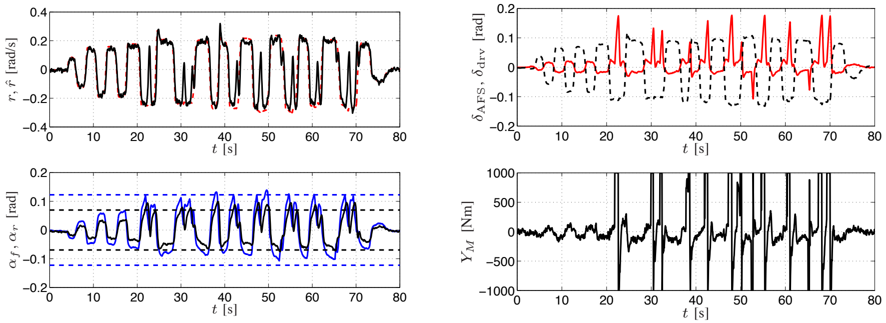
- Simpler solution: assume PWA mode remains constant in prediction
- Design a linear MPC controller for each mode, make it explicit
- 4 linear explicit MPC's are enough (linear/saturation $\times$ front/rear)
- Simulation results:



Experiments

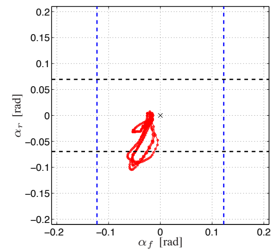
- Experimental results (**slalom test**):

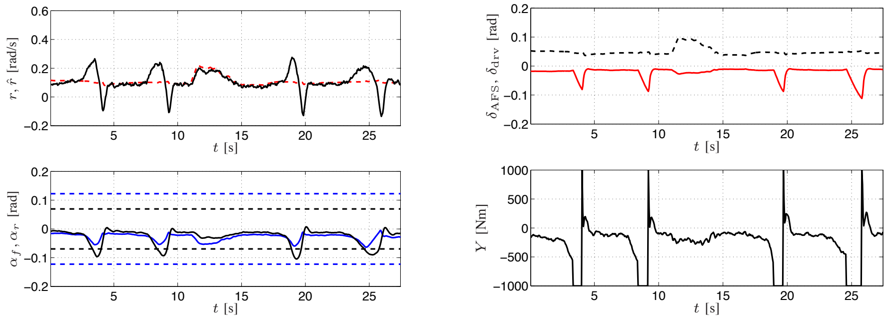




Experiments

- Experimental results (**stability recovery test**):





Experiments

- Experimental results (**double lane change**):