WP4 AA: Automotive Applications

Andrea Balluchi

PARADES
Via San Pantaleo, 66 - 00186 Roma
balluchi@parades.rm.cnr.it

CC 3rd Year Review Meeting
Zurich - March 12, 2005

WP4 AA: Automotive Applications

- **Description of the work:**
  - Task AA.1: Model Construction
    - months 0-24, [ETH, PARADES, SIENA]
  - Task AA.2: Design of Control Algorithms
    - months 6-36, [CWI, ETH, PARADES , SIENA, VERIMAG]
  - Task AA.3: Implementation aware methodologies
    - months 18-36, [PARADES]
  - Task AA.4: Driveline observer design
    - months 0-36 [PARADES]

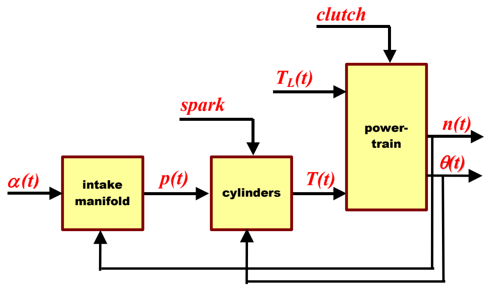
Main Results of the 3rd Year

- Idle speed control
- GDI engine control
- Semiactive suspensions
- Actual engaged gear identification
- Path–following control for bounded–curvature vehicles
- Integrated control-implementation design methodology

Idle Speed Control Case Study

- From a case study to a benchmark
  - very detailed engine hybrid model for controller validation
  - complete specification
  - solved using a polynomial equation approach
- Control algorithm synthesis
  - reachability based approach
  - control-to-facet approach

Idle Speed Control Case Study



Control Problem

Given a value of n_0, Δ and T_L^M, determine whether there exist

- an ignition control *spark* and
- a throttle control $\alpha(t)$

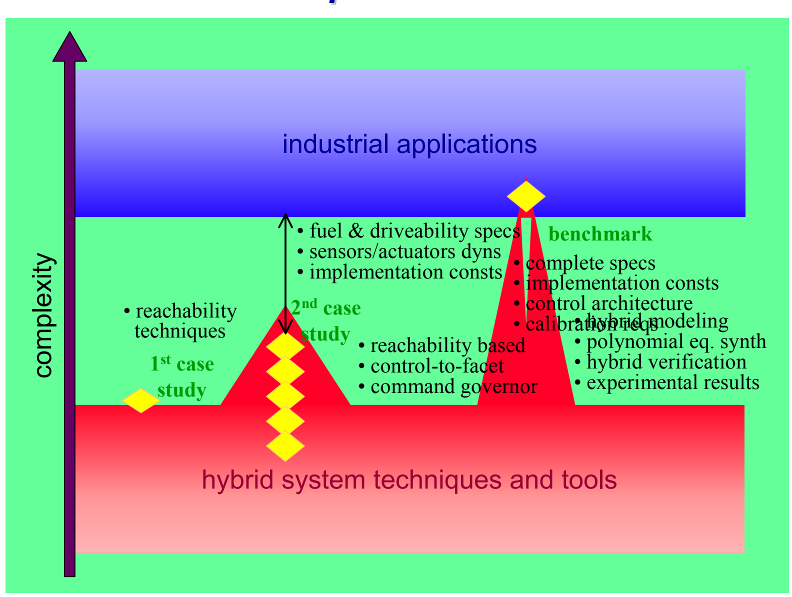
that maintain the crankshaft speed $n(t)$ in the given range $n_0 \pm \Delta$ under

- any driver's action on *clutch* pedal,
- any load torque $T_L(t)$ in $[0, T_L^M]$.

Controls	Time / Value
ignition *spark*	disc / disc
throttle α	cont / cont

Disturbances	Time / Value
clutch *clutch*	disc / disc
load torque T_L	cont / cont

Achievements on idle speed control



Idle speed control - reachability based approach

VERIMAG

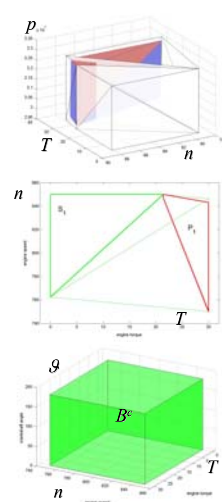
- The maximal controller has been derived from the computation of the maximal invariant set
  - assuming piecewise constant continuous inputs
  - exploiting efficient computation for symbolic constants in d/dt
- From the maximal controller, an optimal controller that minimizes fuel consumption has been extracted
  - using the tool MISER and a discrete-time cycle-based model of the engine
- The performance loss caused by the restriction to the piecewise constant control functions was studied
  - performance loss is quadratic in cycle duration
- Methods for simplifying sets of polyhedra are under investigation
  - compact representation of the maximal invariant set
  - scalability of the hybrid system verification tools as d/dt

Idle speed control - control-to-facet approach

PARADES

CWI

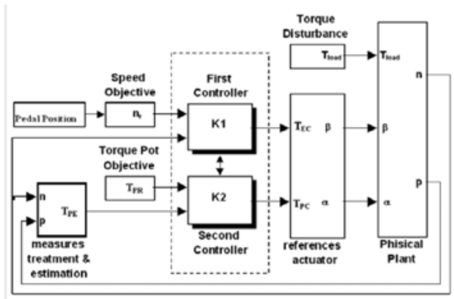
- The control problem was formulated as an affine hybrid system control problem on polytopes and solved by
  - back-stepping and interactive design
  - control for affine systems on polytopes
  - theory of invariant sets
- Open problems
  - synthesis for performance specifications in addition to safety constraints
  - automatic and optimized partitioning of the space in polytopes
  - automatic generation of feedbacks for each polytope



Idle speed control - benchmark polynomial equation approach

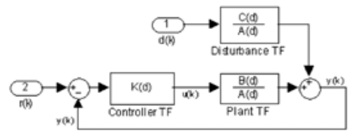
PARADES

- **Desired specification**
  - engine speed constraint
  - driveability requirements
  - fuel consumption



control algorithm architecture

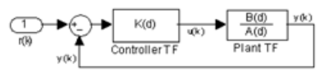
Spark-Advance Reference controller



- dead-beat ripple-free
- fuel minimization / max error const.

$$\min_{W\in\Re^{w}[d]} \|\Delta U\|_{\mathcal{A}_\infty}, \text{ subject to } \|Y\|_{\mathcal{A}_\infty} < \gamma_2$$

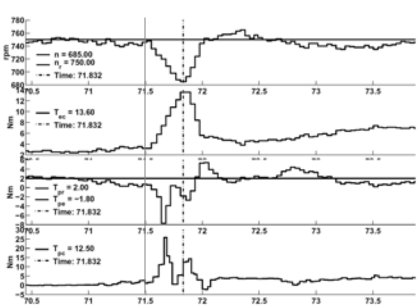
Air-Mass Reference controller

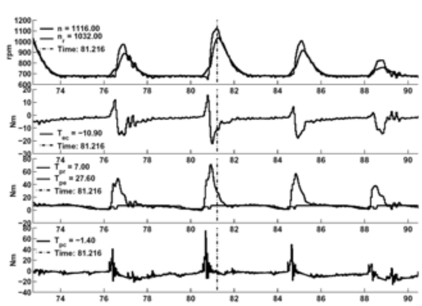


- direct synthesis based on rise time and overshoot requirement

Idle speed control - verification and experimental results

- **Formal verification of robustness with respect to dead center event variations**





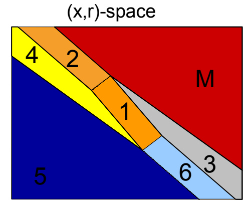
- **Load disturbance step response**
  - 50% undershoot reduction wrt PID/LQ
  - very small overshoot
  - no spark control saturation
- **Tracking of an oscillating reference**
  - PID/LQ large undershoot, oscillation
  - very good tracking performances
  - no undershoot, no speed fluctuations
  - low fuel consumption

MPC Controller Synthesis Approach

- **MPC control design handles all performance specs and constraints in a natural and direct way**
- **Piecewise affine MPC controllers can be synthesized, off-line, and implemented as look-up tables of linear gains**

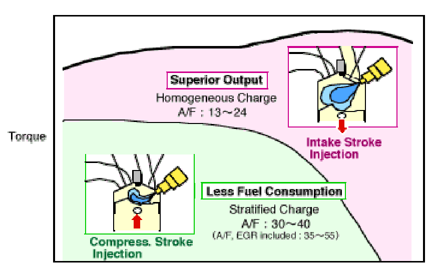
$$u(x,r) = \begin{cases} F_1 x + E_1 r + g_1 & \text{if } H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \vdots \\ F_M x + E_M r + g_M & \text{if } H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$

(x,r)-space



- **Matlab tools are available to assist the whole design process**
  http://www.dii.unisi.it/hybrid/toolbox

GDI Engine Control



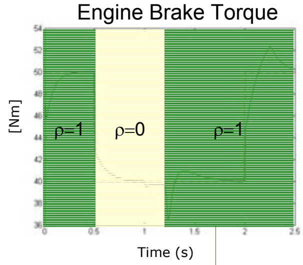
- **Homogeneous charge (ρ=1): fuel is injected in the intake stroke**
  - **stoichiometric injection: A/F=14.64**
- **Stratified charge (ρ =0): fuel is injected in the compression stroke**
  - **lean mixtures: A/F > 14.64**

Dynamic equations are nonlinear
Dynamics and constraints depend on regime ρ !

→ **Hybrid Model**

GDI Engine Control - MPC Controller

Engine Brake Torque



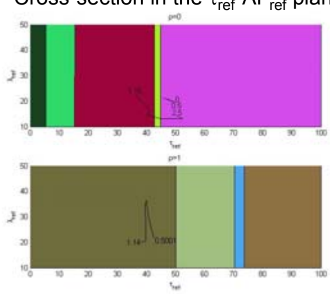
On-line simulations:

- Control horizon N=1;
- Simulation time T=2.5 s;
- Sampling time Ts=10 ms;
- CPU time $\simeq$ 8 s;
(on an Intel Centrino 1.2 GHz, 640 Mb RAM with Cplex 9.01)

→ $\simeq$ 32 ms per time step

Not directly implementable !

Cross-section in the τ_{ref}-AF_{ref} plane



Explicit simulations:

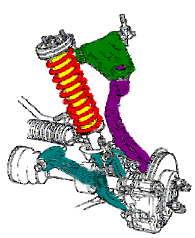
- Control horizon N=1;
- 9 Parameters;
- CPU time $\simeq$ 0.52 s

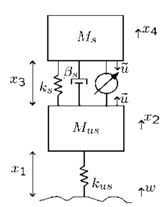
→ **75** Partitions

$\simeq$ 2 ms per time step

Experimental results with fast prototyping

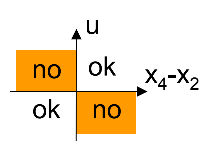
Semiactive Suspensions





quarter-car linear model

1) Passivity condition: $\tilde{u}(x_4 - x_2) \geq 0$



2) Max dissipation power:

$$0 \leq \tilde{u}(x_4 - x_2) \leq (2 \cdot 25.5 \cdot \omega_s)(x_4 - x_2)^2$$

3) Saturation: $\tilde{u}_{\min} \leq \tilde{u} \leq \tilde{u}_{\max}$

(1), (2) are nonlinear & nonconvex physical constraints → **Hybrid Model**

Semiactive Suspensions - 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;
} } }
```

- **A Matlab MLD object is generated using the Hybrid Toolbox for simulation, analysis, and control design.**

Semiactive Suspensions - Explicit MPC

- **Horizon N=1: same as Clipped-LQR !**
- **For increasing N: better closed-loop performance**

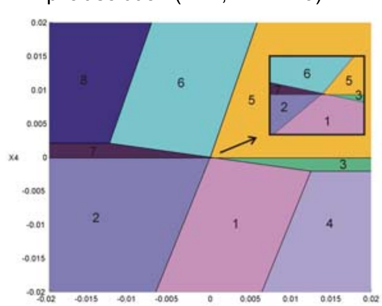
Performance Index

N	MPC	Clipped-LQR
1	0.7556	0.7556
5	0.7518	
10	0.7252	
15	0.6590	
20	0.6562	
30	0.6192	
40	0.6206	

Same Cost Value !

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

Explicit solution (N=1, x1=x2=0):



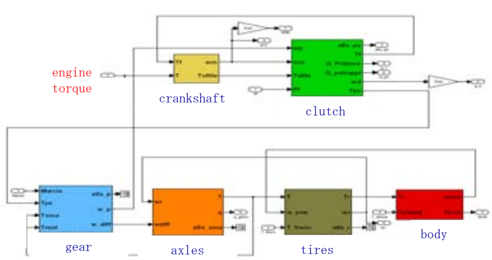
- 4 parameters, 8 regions
- Regions #1, #6: ExMPC=Clipped-LQR;
- Regions #3, #7: max dissipation power;
- Regions #4, #8: saturated constraint.

Actual Engaged Gear Identification

PARADES

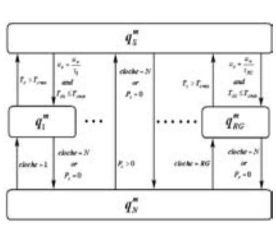
- **Main motivations**
  - driveability and tailpipe emission control
- **Automotive driveline modeling**
  - very detailed nonlinear hybrid model
  - abstracted and reduced model for synthesis
- **Identification algorithm based on hybrid observers**
- **Validation**
  - with detailed hybrid nonlinear model
  - with experimental data provided by Magneti Marelli Powertrain
- **Robustness analysis**

Driveline modeling



detailed hybrid model:

6048 discrete states
12 continuous states



$q^m{}_N$: $$\dot{\omega}_e(t) = -\frac{b_e}{j_e}\omega_e(t) + \frac{1}{j_e}T_e(t)$$

$q^m{}_S$: $$\dot{\omega}_e(t) = -\frac{b_e}{j_e}\omega_e(t) + \frac{1}{j_e}T_e(t) - \frac{1}{j_e}T_c(t)$$

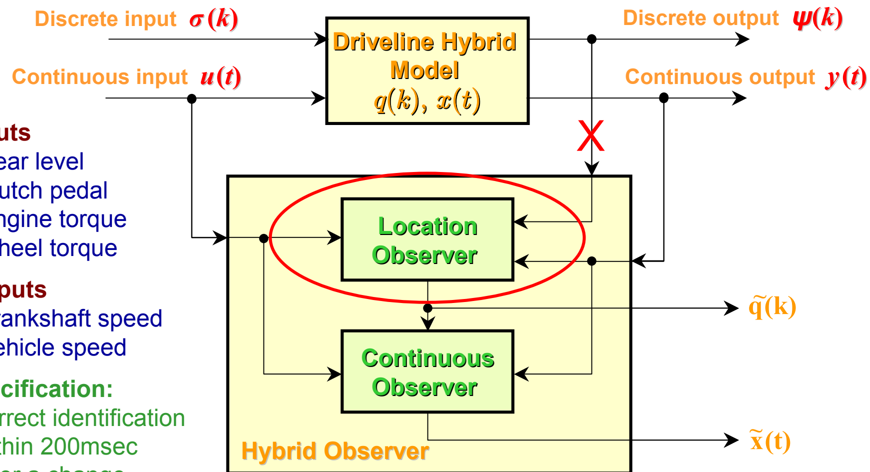
$q^m{}_i$: $$\dot{x}(t) = A_i x(t) + B_i u(t)$$
$$y(t) = C_i x(t)$$
$$x = \begin{pmatrix} \alpha \\ \omega_e \\ \omega_w \end{pmatrix}, \quad u = \begin{pmatrix} T_e \\ T_w \end{pmatrix}, \quad y = \begin{pmatrix} \omega_e \\ \omega_w \end{pmatrix}$$

reduced hybrid model:

8 discrete states
- 6 gear
- 2 clutch

3 continuous states
- crankshaft speed
- vehicle speed
- driveline torsion angle

A hybrid observer approach to actual engaged gear identification

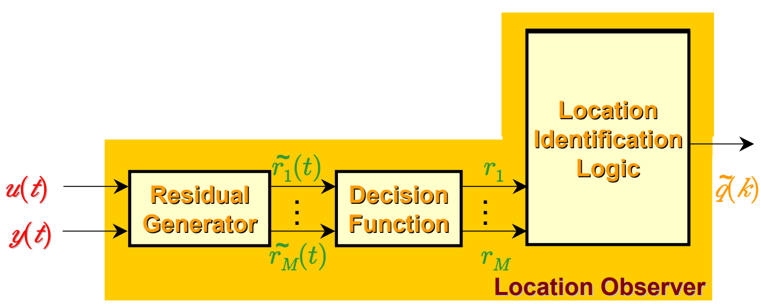


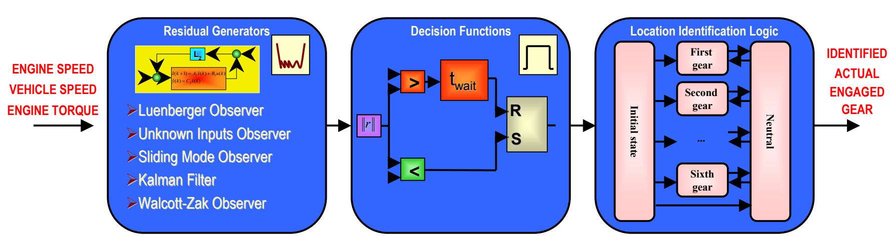
inputs
- gear level
- clutch pedal
- engine torque
- wheel torque

outputs
- crankshaft speed
- vehicle speed

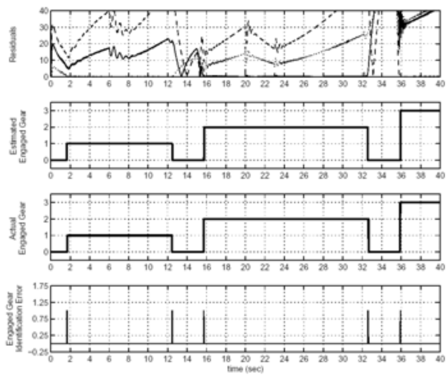
specification:
correct identification within 200msec after a change

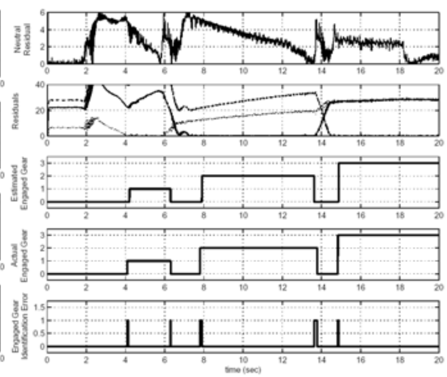
Identification algorithm





Algorithm validation

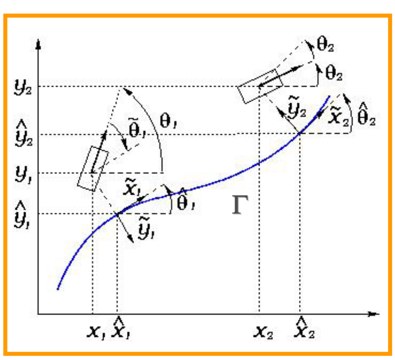


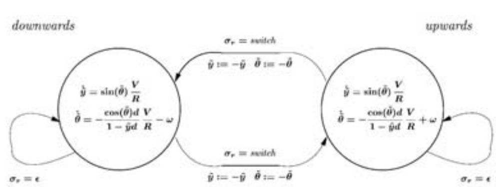


- Simulation results obtained with the very detailed hybrid driveline model
- Experimental results obtained in Magneti-Marelli with an Opel Astra

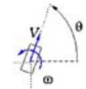
Path–following Control for Bounded–curvature Vehicles

PARADES



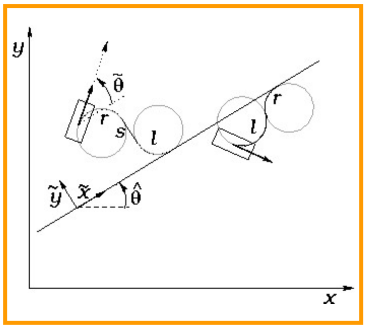


$|\omega(t)| \leq \frac{V}{R}$



- Problem: steer and stabilize the vehicle on a generic path
  - only curvature sign available on-line (not amplitude)
  - lateral position and orientation error measurable
- **Hybrid vehicle model based on Frenet's frame**

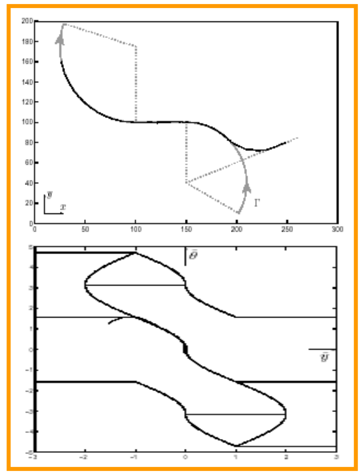
Path–following control - hybrid algorithm



based on optimal control to reach a line

- *go_straight* where $\omega = 0$, if $(\tilde{y}, \tilde{\theta}) \in \Omega^0$
- *turn_right* where $\omega = -\frac{V}{R}$, if $(\tilde{y}, \tilde{\theta}) \in \Omega^-$
- *turn_left* where $\omega = +\frac{V}{R}$, if $(\tilde{y}, \tilde{\theta}) \in \Omega^+$

Robust stability proved by formal verification

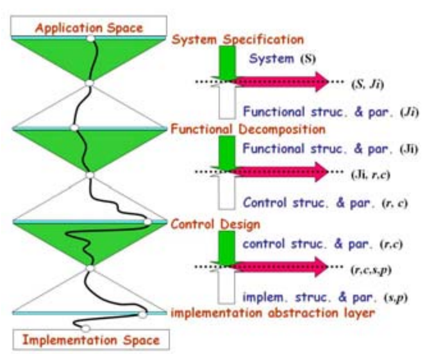


experimental results with mobile robot

Integrated control-implementation design methodology

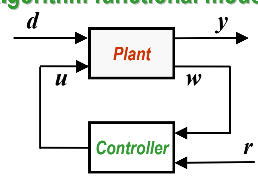
PARADES

- In derivative design, algorithms are selected from available libraries
- Validation of candidate solutions is obtained by plant/controller parameter exploration
- Using an implementation abstraction layer
  - ONLY implementation details that affect closed-loop performance are exported
- Integrated control-implementation design is formalized as
  - a refinement of the controller into instances of the implementation abstraction layer
- Validation of the implementation platform is obtained by implementation parameters exploration



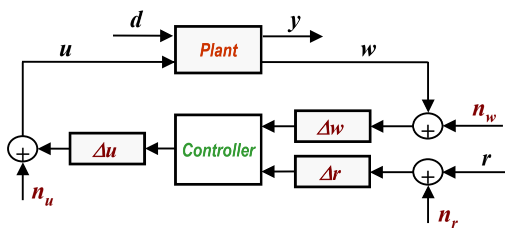
Control algorithm validation models

- control algorithm functional model



- y : performance outputs
- d : disturbances
  - measurable / unmeasurable
- r : reference signals
  - from other operations or commands
- w : feedback / feedforward signals
- u : control signals

- control algorithm implementation abstract model



- Δu, Δr, Δw : time-domain perturbations
  - control loop delays
  - sample & hold
  - etc.
- n_u, n_r, n_w :value-domain perturbations
  - quantization error
  - computation imprecision
  - etc.

Randomized techniques for controller and implementation exploration

- A tool has been developed in Matlab/Simulink for
  - functional and
  - implementation platform

exploration.

- Validation algorithms are based on randomized techniques
  - control and implementation parameters described in a stochastic framework
- Quantifier elimination is used to abstract control parameters and explore different implementations
- Applied to an entire algorithm chain for fuel injection control