On the design process of automotive control systems

Andrea Balluchi
Alberto Sangiovanni Vincentelli

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

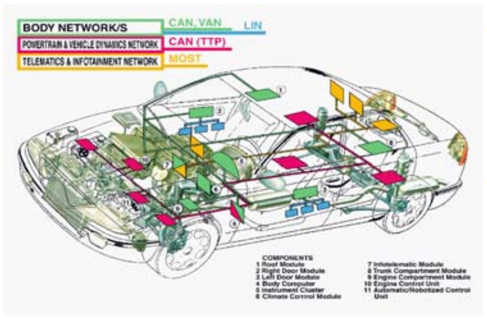
CC 3rd Year Review Meeting
Zurich - March 12, 2005

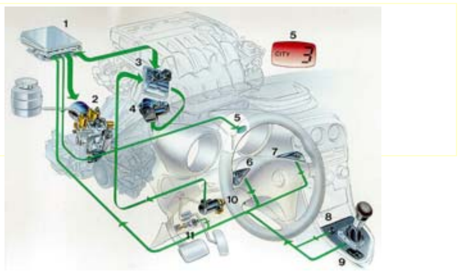
Project IST-2001-33520: CC - Computation and Control

Automotive industry scenario

- Customers are more demanding in terms of
  - quality, style and functionality of cars
- Manufacturers have to redesign their products more frequently
  - reducing the design cycle
  - introducing innovation in each renewal
- To master the complexity of design and manufacturing
  - upstream validation of industrial feasibility of new products
  - rationalization of all production system design activities
  - accurate sizing of manufacturing system

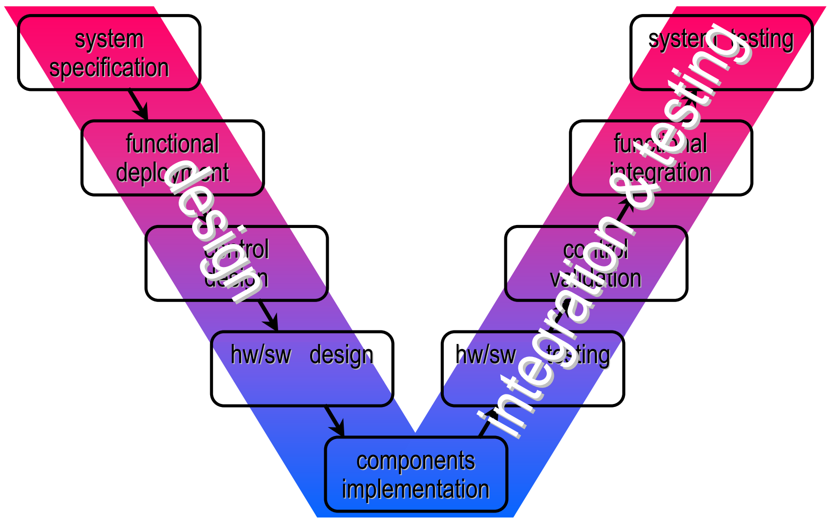
Automotive networked control system



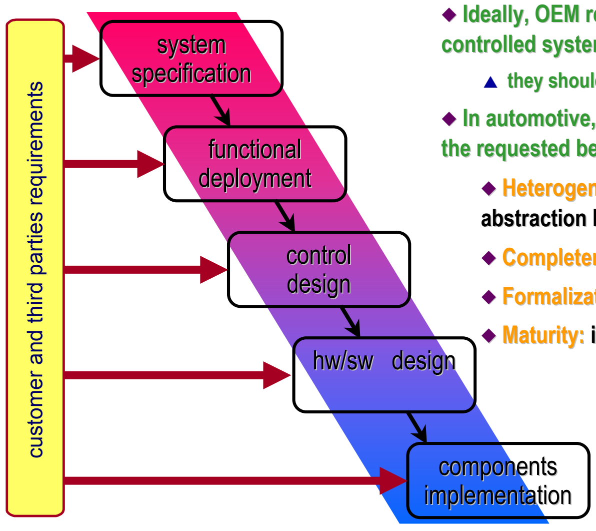


- The pressure of competitiveness is even higher for control system development, since more than 80% of innovation is in electronics
- In today cars, the electronic control system is a networked system
  - with more than 80 interconnected ECUs
  - e.g. engine control, gear box control, ABS, traction control, dashboard control, vehicle dynamic control (VDC)

Automotive industry design process

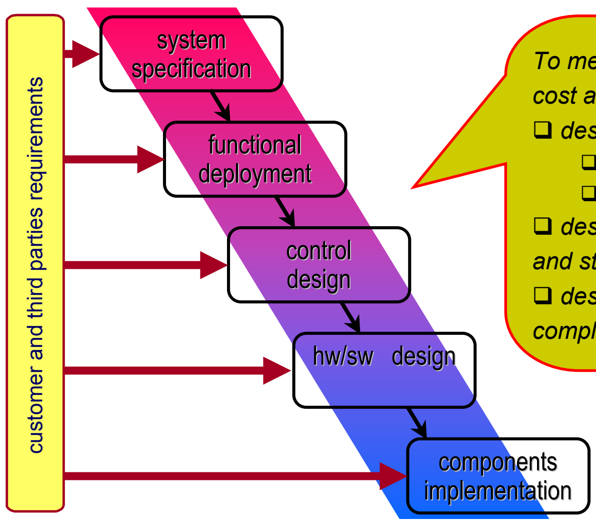


Design requirements for automotive controllers



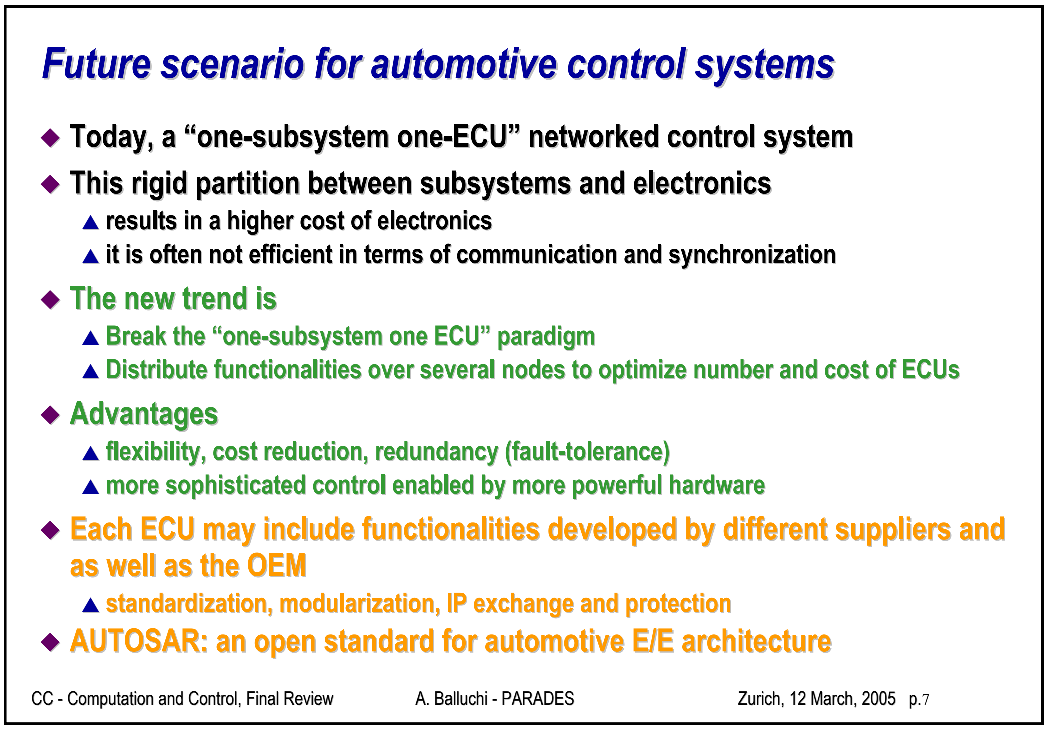
- Ideally, OEM requirements should define what controlled system must do
  - they should be given at system level
- In automotive, requirements often define, in part, how the requested behavior is implemented
  - Heterogeneity: at the same and at different abstraction levels
  - Completeness: not complete at the top level
  - Formalization: most of them not formally specified
  - Maturity: initially only part of them available

Design requirements for automotive controllers



To meet very tight constraints on cost and time-to-market:

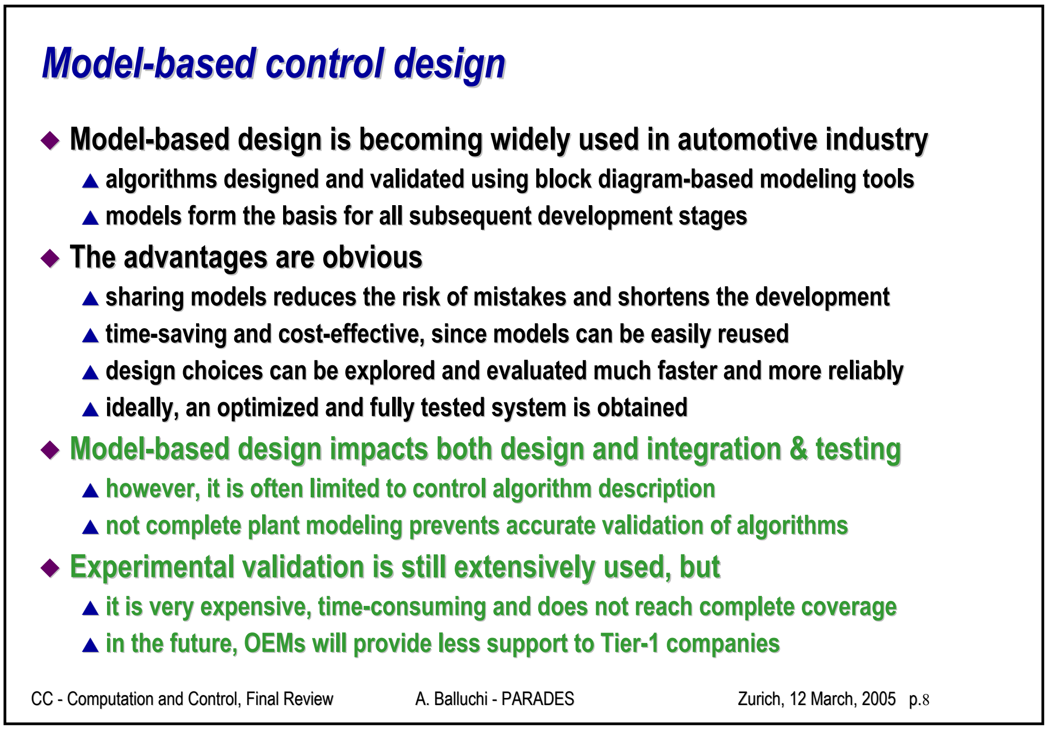
- ❑ *design methodology has to*
  - ❑ *support and foster reuse*
  - ❑ *be based on derivative design*
- ❑ *design chain has to be integrated and standardized*
- ❑ *design tools have to meet complementary & interoperability*



Future scenario for automotive control systems

- Today, a "one-subsystem one-ECU" networked control system
- This rigid partition between subsystems and electronics
  - results in a higher cost of electronics
  - it is often not efficient in terms of communication and synchronization
- The new trend is
  - Break the "one-subsystem one ECU" paradigm
  - Distribute functionalities over several nodes to optimize number and cost of ECUs
- Advantages
  - flexibility, cost reduction, redundancy (fault-tolerance)
  - more sophisticated control enabled by more powerful hardware
- Each ECU may include functionalities developed by different suppliers and as well as the OEM
  - standardization, modularization, IP exchange and protection
- AUTOSAR: an open standard for automotive E/E architecture

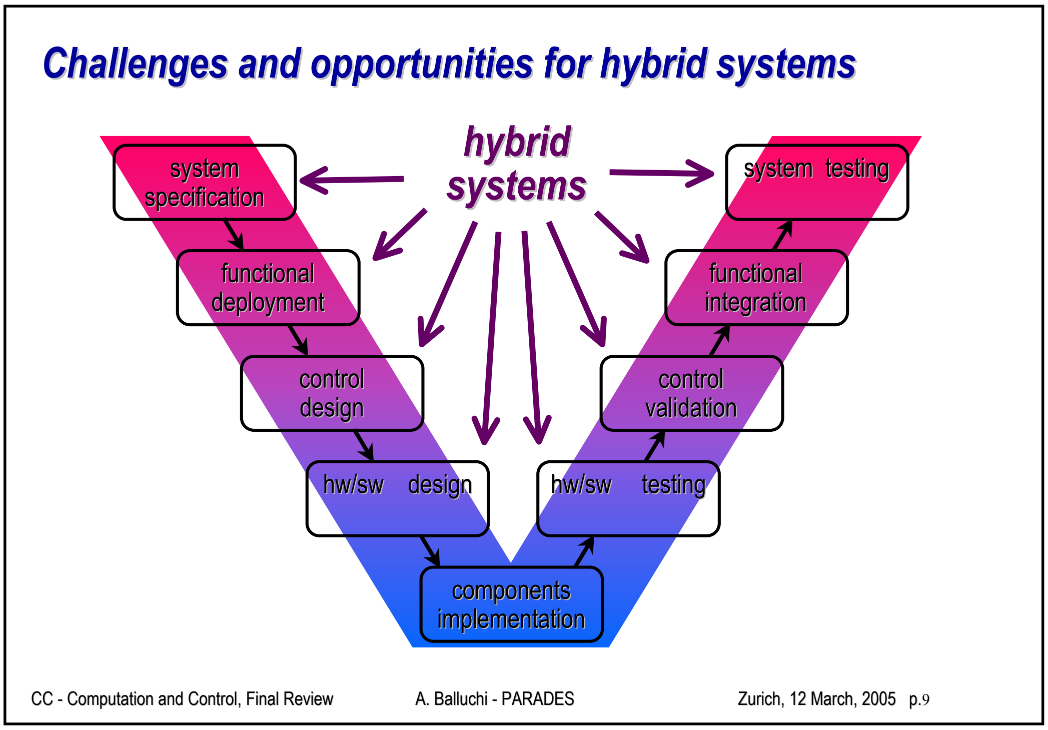
CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.7



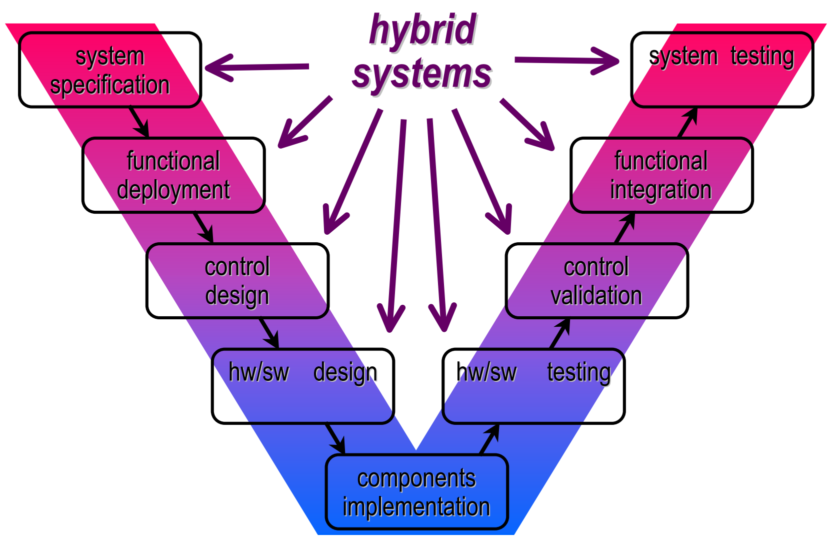
Model-based control design

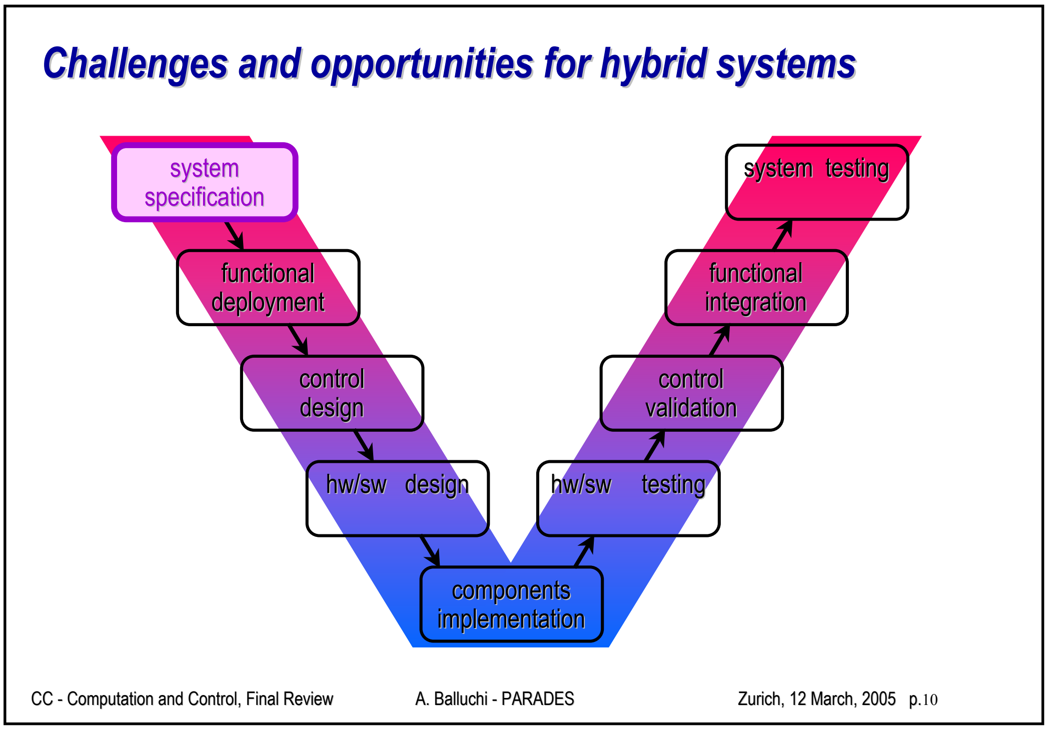
- Model-based design is becoming widely used in automotive industry
  - algorithms designed and validated using block diagram-based modeling tools
  - models form the basis for all subsequent development stages
- The advantages are obvious
  - sharing models reduces the risk of mistakes and shortens the development
  - time-saving and cost-effective, since models can be easily reused
  - design choices can be explored and evaluated much faster and more reliably
  - ideally, an optimized and fully tested system is obtained
- Model-based design impacts both design and integration & testing
  - however, it is often limited to control algorithm description
  - not complete plant modeling prevents accurate validation of algorithms
- Experimental validation is still extensively used, but
  - it is very expensive, time-consuming and does not reach complete coverage
  - in the future, OEMs will provide less support to Tier-1 companies

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.8

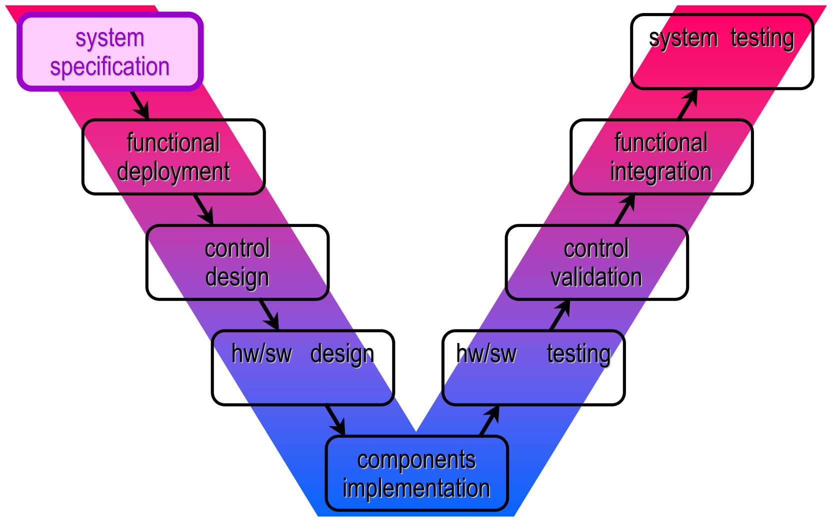


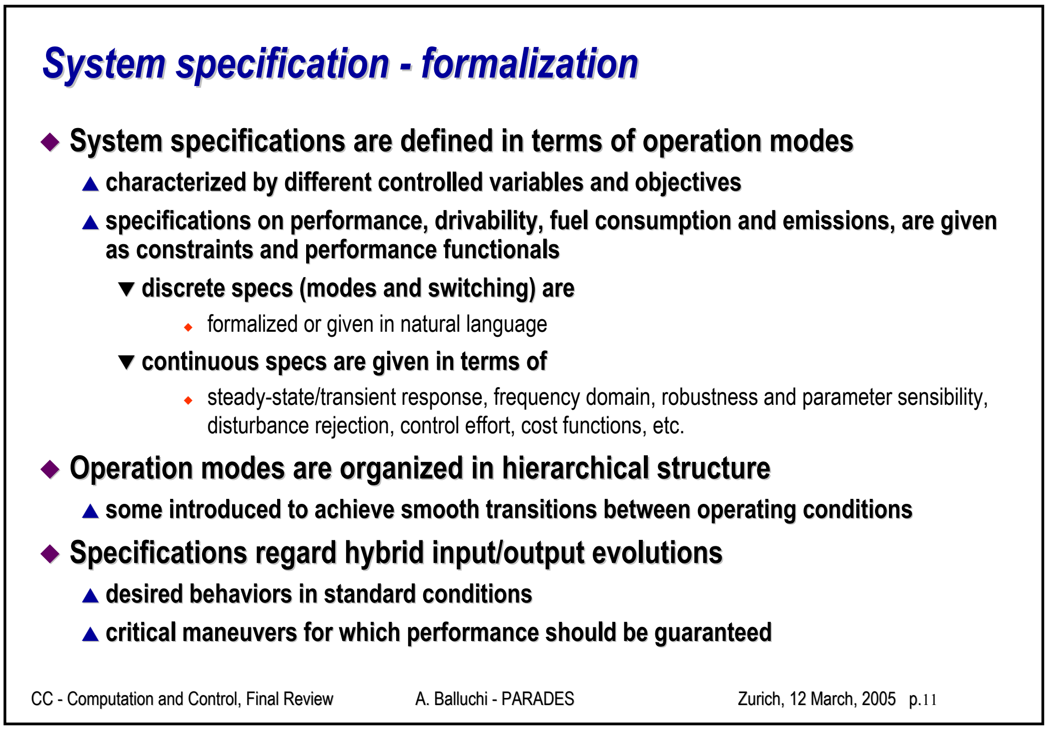
Challenges and opportunities for hybrid systems





Challenges and opportunities for hybrid systems

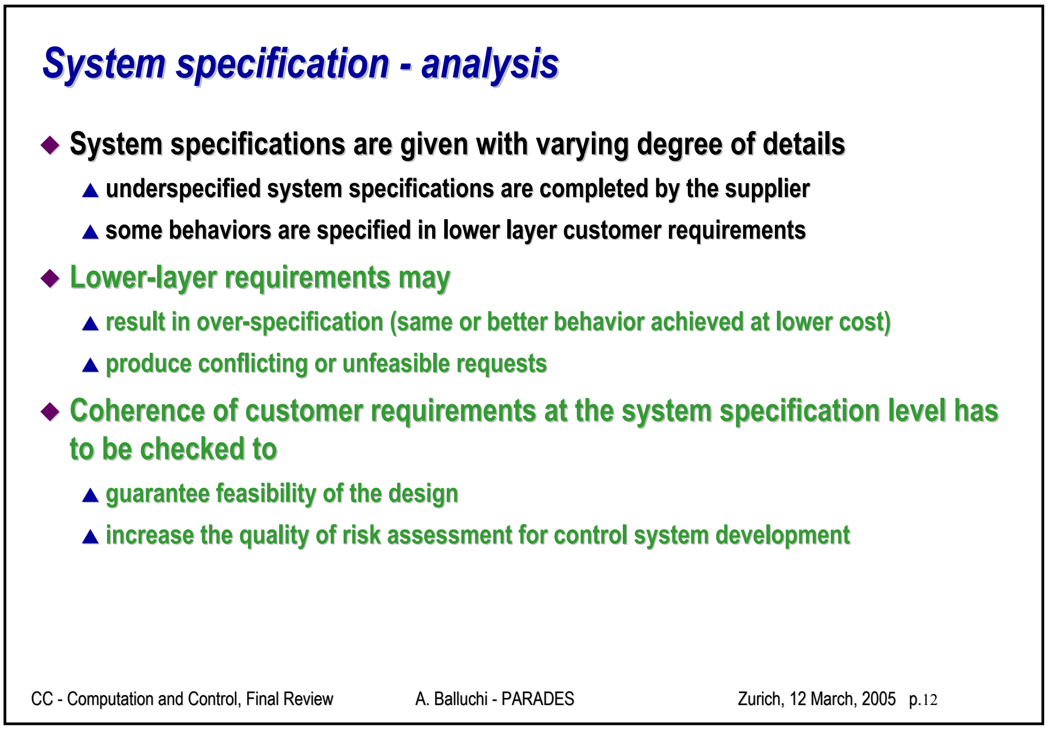




System specification - formalization

- **System specifications are defined in terms of operation modes**
  - **characterized by different controlled variables and objectives**
  - **specifications on performance, drivability, fuel consumption and emissions, are given as constraints and performance functionals**
    - **discrete specs (modes and switching) are**
      - formalized or given in natural language
    - **continuous specs are given in terms of**
      - steady-state/transient response, frequency domain, robustness and parameter sensibility, disturbance rejection, control effort, cost functions, etc.
- **Operation modes are organized in hierarchical structure**
  - **some introduced to achieve smooth transitions between operating conditions**
- **Specifications regard hybrid input/output evolutions**
  - **desired behaviors in standard conditions**
  - **critical maneuvers for which performance should be guaranteed**

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.11



System specification - analysis

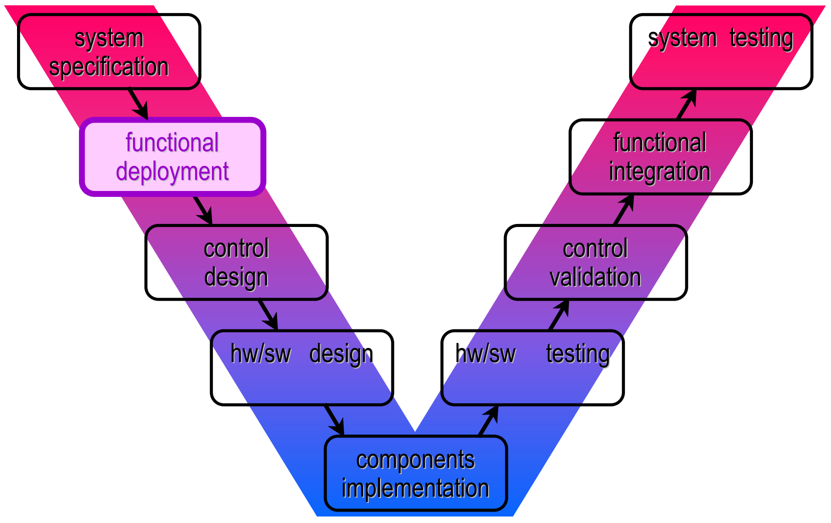
- **System specifications are given with varying degree of details**
  - **underspecified system specifications are completed by the supplier**
  - **some behaviors are specified in lower layer customer requirements**
- **Lower-layer requirements may**
  - **result in over-specification (same or better behavior achieved at lower cost)**
  - **produce conflicting or unfeasible requests**
- **Coherence of customer requirements at the system specification level has to be checked to**
  - **guarantee feasibility of the design**
  - **increase the quality of risk assessment for control system development**

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.12

System specification

- Since the clear trend is towards model-based design, specs in executable form are attractive
- Hybrid formalisms for system specification description
- Supporting tools for system specification modeling
  - interoperability with: requirement management tools and system engineering tools for control algorithm design, validation and verification
- Abstraction techniques for mapping lower layers specs to upper layers
  - to achieve completeness of the description at system specification level
- Methodologies and tools for validation of system specification
  - feasibility, conflicting requirements, etc
  - quantitative techniques to support risk assessment

Challenges and opportunities for hybrid systems



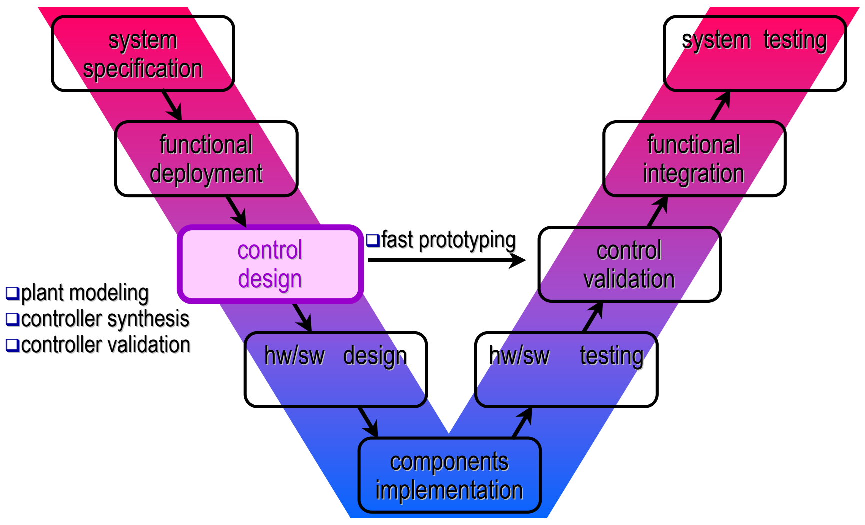
Functional deployment

- The system is viewed as a collection of interacting subsystems
- For each subsystem (or functional component)
  - a desired behavior is defined in order to meet system specification
  - the architecture of control algorithms is defined
  - specifications for the design each control algorithm are established
- Input/output connections (as well as sensors and actuators) are often defined in the customer requirements
- Functional deployment is highly guided by the experience of system engineers, with little support of methodologies and tools
- Traceability of customer requirements to functional deployment results is a key issue

Functional deployment

- Hybrid formalisms and modeling tools for the description of
  - desired behavior of each functional component
  - architecture of control algorithms for each functional component
  - desired requirements for each control algorithm
- Methodologies and tools for evaluation and validation with respect to system specification of
  - desired behavior of each functional component
  - desired requirements for each control algorithm

Challenges and opportunities for hybrid systems



Control design

- **Plant modeling**
  - **model development**
  - **identification**
  - **validation**
- **Controller synthesis**
  - **plant model and specifications analysis**
  - **algorithms development**
  - **controller validation**
- **Fast prototyping**

Derivative design approach

- **Every two-three years, a new generation of products is designed**
  - **Product generations are conceived to accommodate the specification of all customers for the next years**
- **For each commitment, the ECU is obtained by derivation from the current generation**
  - **Reuse is extensively employed to minimize cost and development time**

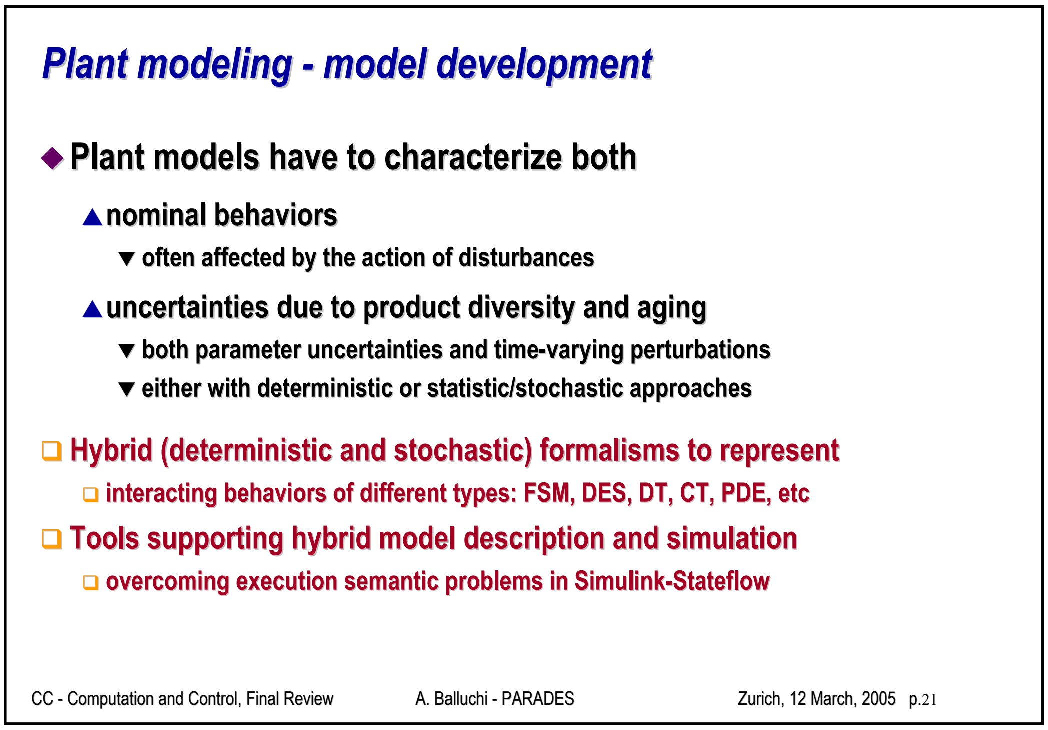
Control design

- **Plant modeling**
  - **model development**
  - **identification**
  - **validation**
- Controller synthesis
  - plant model and specifications analysis
  - algorithms development
  - controller validation
- Fast prototyping

Plant modeling - model development

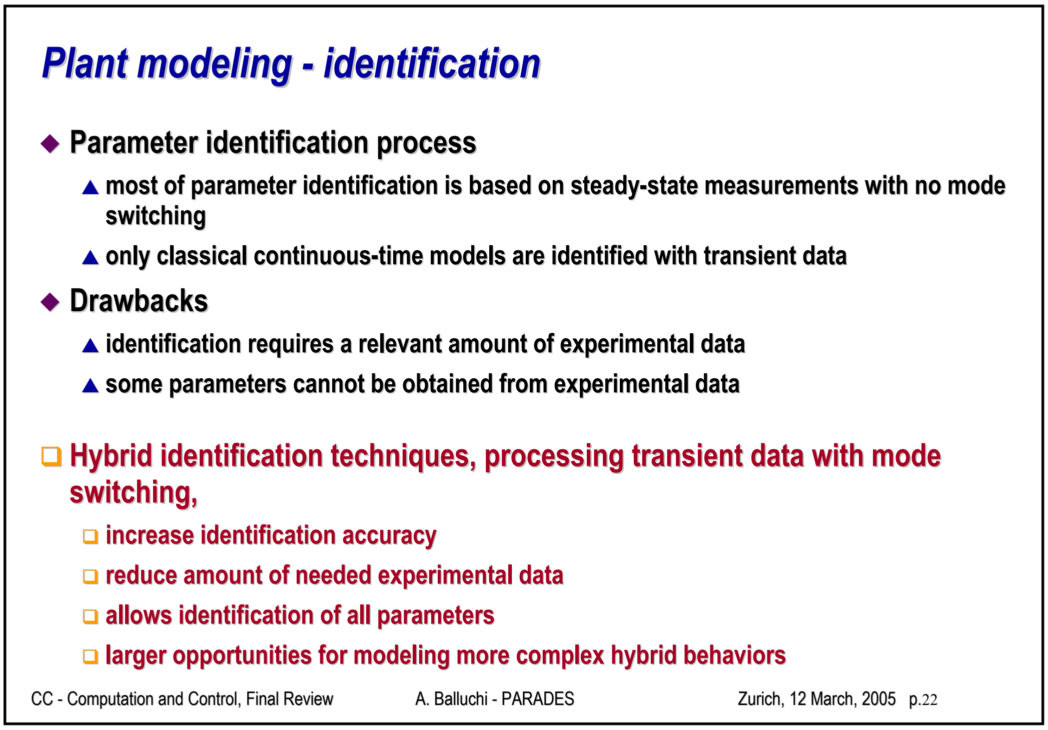
	DT	CT
DV	🟦	🟪
CV	🟫	🟥

- **Hybrid behaviors in automotive subsystems**
  - **internal combustion engine**
    - **4-stroke engine cycle (FSM + DES + CT)**
      - inputs: spark ignition; injected fuel; air charge; exhaust gas concentration in air charge; engine speed;
      - outputs: engine torque; crankshaft events; engine temperature; air-to-fuel ratio; engine exhaust gas;
    - **fuel injection (FSM + CT)**
      - inputs: fuel injection signal (rail pressure regulator command - DI);
      - outputs: injected fuel (rail pressure;fuel temperature - DI);
    - **spark ignition (FSM)**
      - inputs: ignition coil command; spark command;
      - outputs: spark ignition;
    - **air dynamics (CT)**
      - inputs: throttle valve motor command; EGR valve command; VGT command;
      - outputs: throttle valve position; intake manifold temperature and pressure; air flow rate; air charge; exhaust gas concentration in air charge;
- **Human-factors and man-machine interface issues**



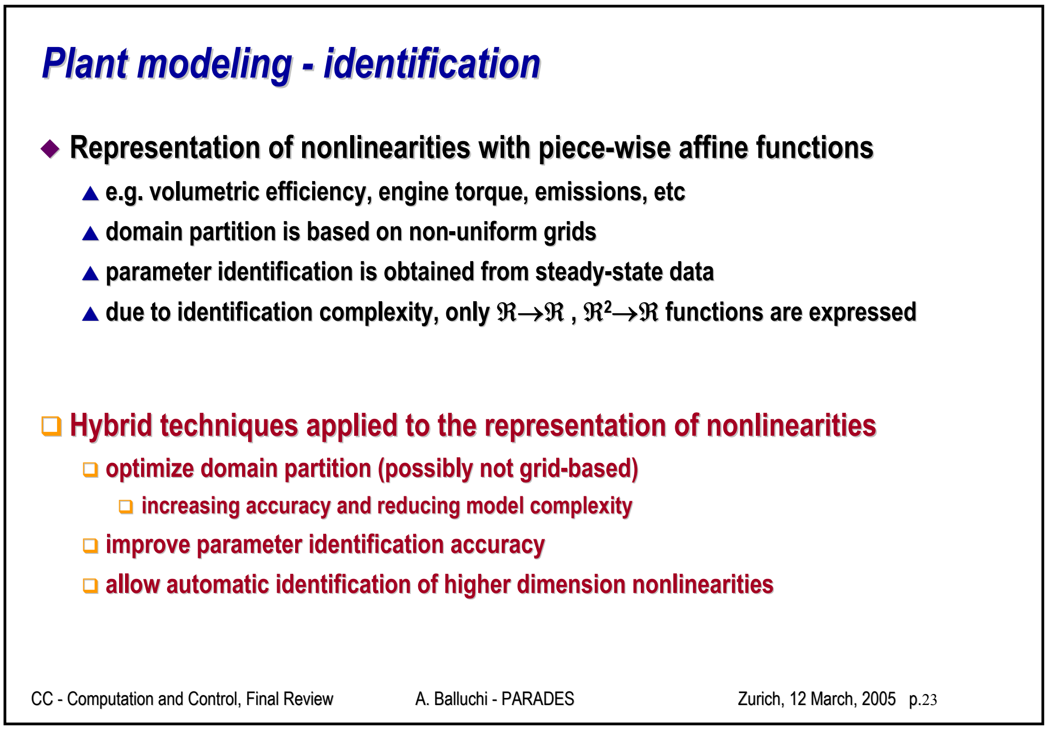
Plant modeling - model development

- Plant models have to characterize both
  - nominal behaviors
    - often affected by the action of disturbances
  - uncertainties due to product diversity and aging
    - both parameter uncertainties and time-varying perturbations
    - either with deterministic or statistic/stochastic approaches
- Hybrid (deterministic and stochastic) formalisms to represent
  - interacting behaviors of different types: FSM, DES, DT, CT, PDE, etc
- Tools supporting hybrid model description and simulation
  - overcoming execution semantic problems in Simulink-Stateflow



Plant modeling - identification

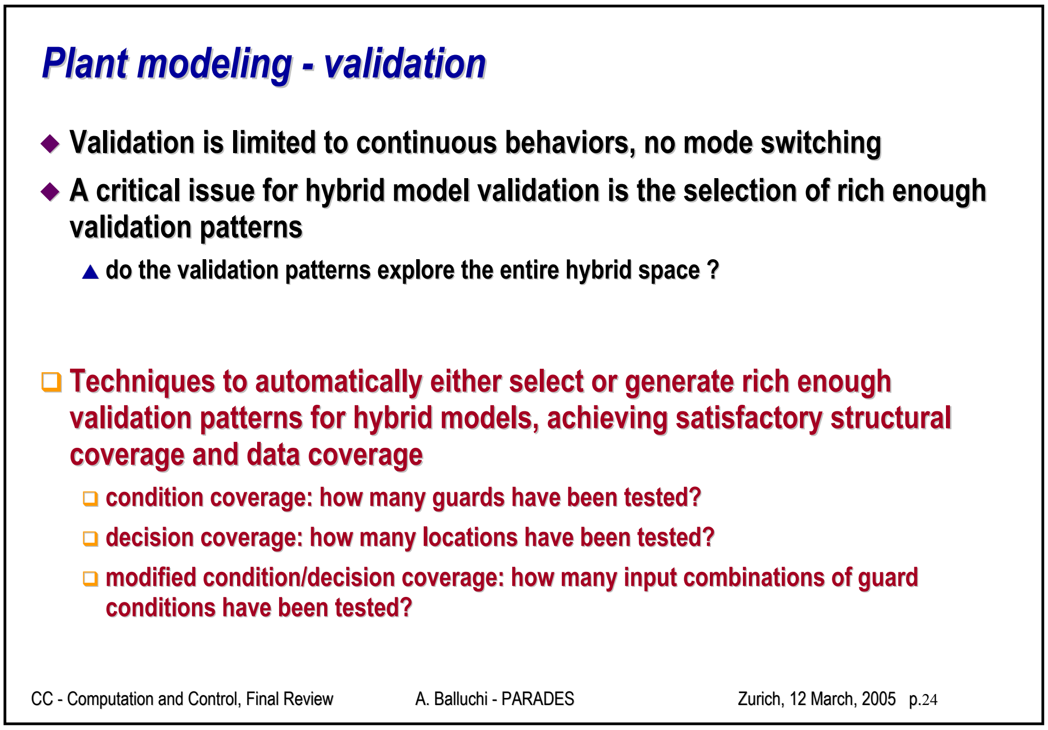
- Parameter identification process
  - most of parameter identification is based on steady-state measurements with no mode switching
  - only classical continuous-time models are identified with transient data
- Drawbacks
  - identification requires a relevant amount of experimental data
  - some parameters cannot be obtained from experimental data
- Hybrid identification techniques, processing transient data with mode switching,
  - increase identification accuracy
  - reduce amount of needed experimental data
  - allows identification of all parameters
  - larger opportunities for modeling more complex hybrid behaviors



Plant modeling - identification

- Representation of nonlinearities with piece-wise affine functions
  - e.g. volumetric efficiency, engine torque, emissions, etc
  - domain partition is based on non-uniform grids
  - parameter identification is obtained from steady-state data
  - due to identification complexity, only $\Re\rightarrow\Re$, $\Re^2\rightarrow\Re$ functions are expressed

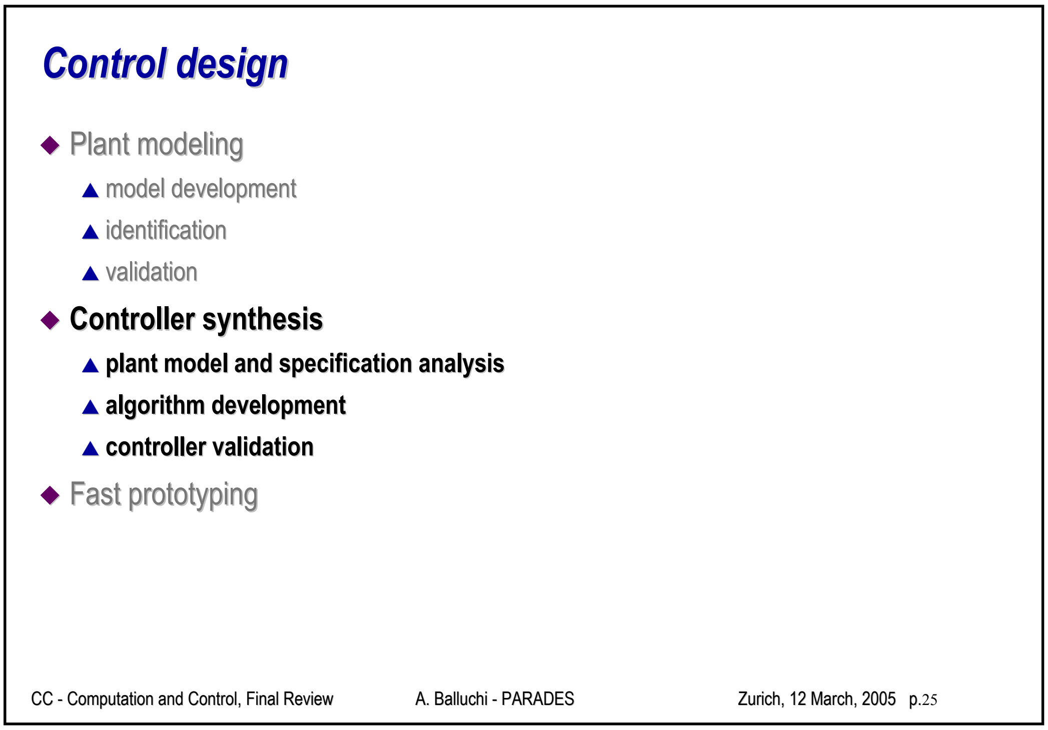
- Hybrid techniques applied to the representation of nonlinearities
  - optimize domain partition (possibly not grid-based)
    - increasing accuracy and reducing model complexity
  - improve parameter identification accuracy
  - allow automatic identification of higher dimension nonlinearities



Plant modeling - validation

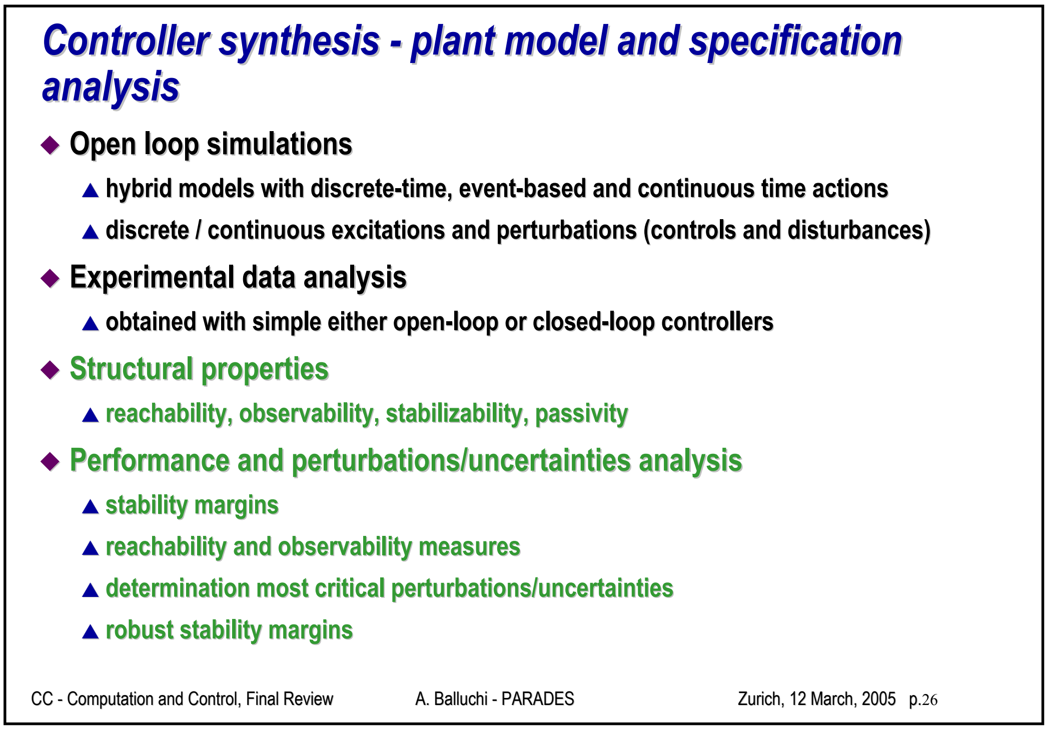
- Validation is limited to continuous behaviors, no mode switching
- A critical issue for hybrid model validation is the selection of rich enough validation patterns
  - do the validation patterns explore the entire hybrid space ?

- Techniques to automatically either select or generate rich enough validation patterns for hybrid models, achieving satisfactory structural coverage and data coverage
  - condition coverage: how many guards have been tested?
  - decision coverage: how many locations have been tested?
  - modified condition/decision coverage: how many input combinations of guard conditions have been tested?



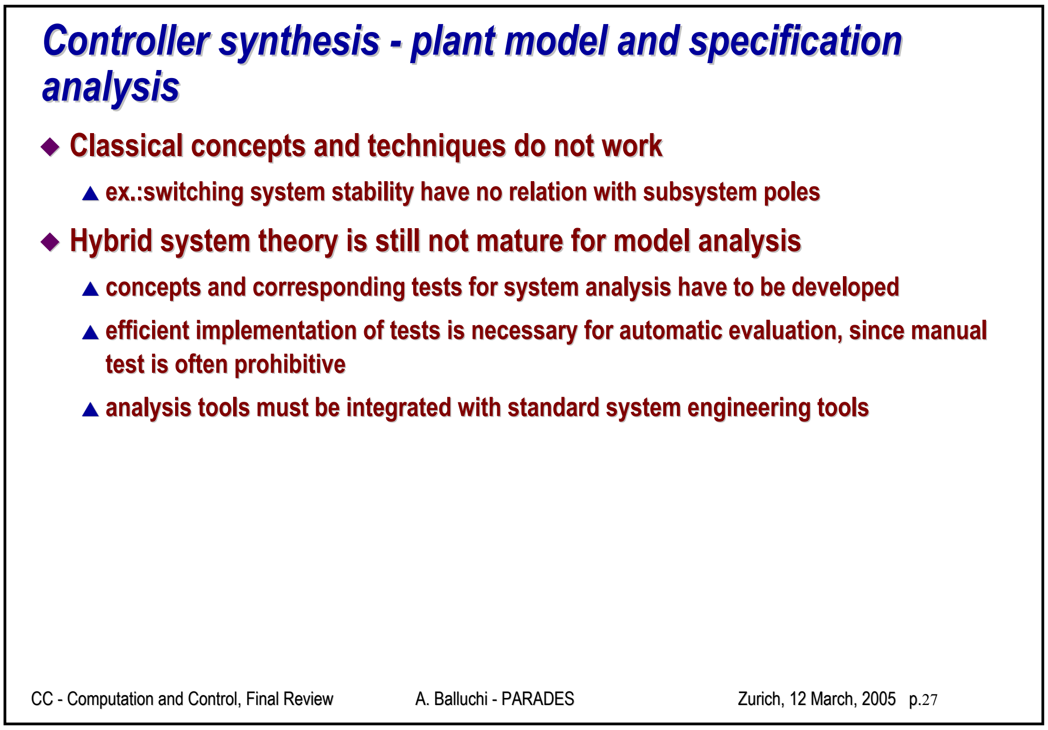
Control design

- Plant modeling
  - model development
  - identification
  - validation
- **Controller synthesis**
  - **plant model and specification analysis**
  - **algorithm development**
  - **controller validation**
- Fast prototyping



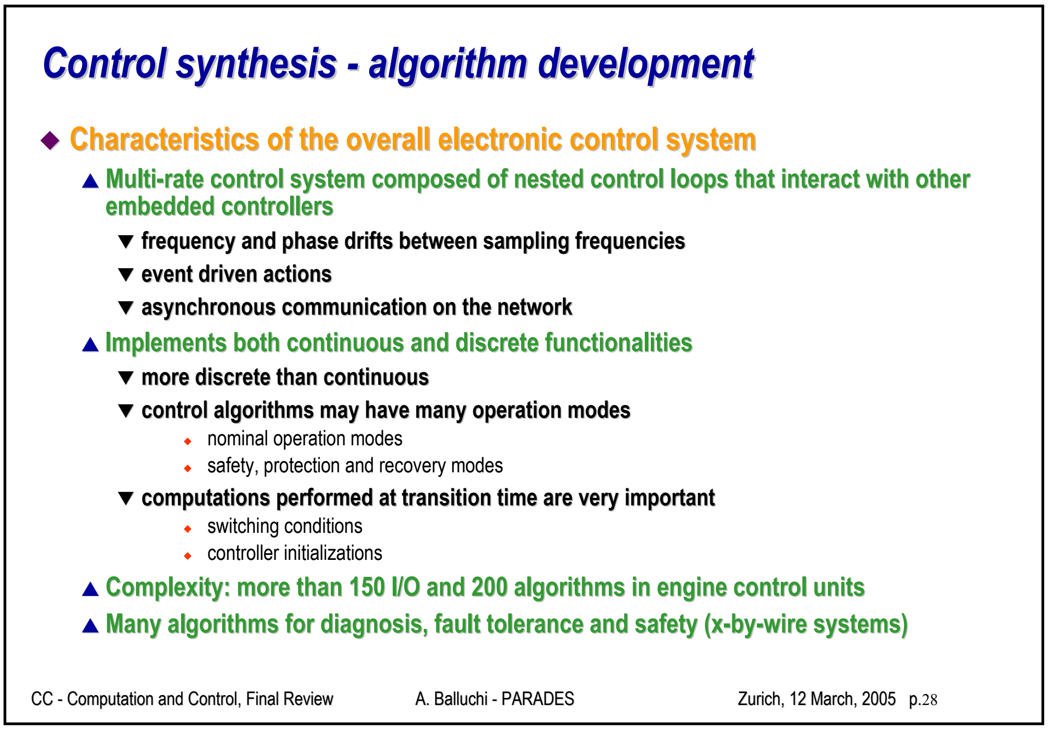
Controller synthesis - plant model and specification analysis

- **Open loop simulations**
  - **hybrid models with discrete-time, event-based and continuous time actions**
  - **discrete / continuous excitations and perturbations (controls and disturbances)**
- **Experimental data analysis**
  - **obtained with simple either open-loop or closed-loop controllers**
- **Structural properties**
  - **reachability, observability, stabilizability, passivity**
- **Performance and perturbations/uncertainties analysis**
  - **stability margins**
  - **reachability and observability measures**
  - **determination most critical perturbations/uncertainties**
  - **robust stability margins**



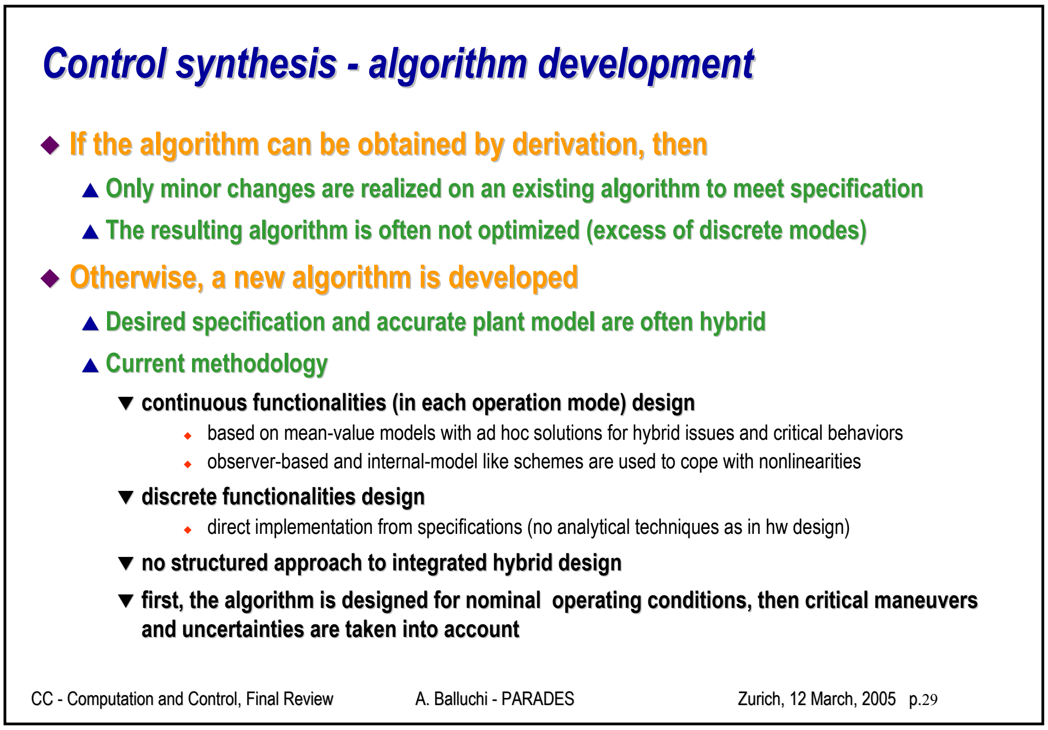
Controller synthesis - plant model and specification analysis

- Classical concepts and techniques do not work
  - ex.:switching system stability have no relation with subsystem poles
- Hybrid system theory is still not mature for model analysis
  - concepts and corresponding tests for system analysis have to be developed
  - efficient implementation of tests is necessary for automatic evaluation, since manual test is often prohibitive
  - analysis tools must be integrated with standard system engineering tools



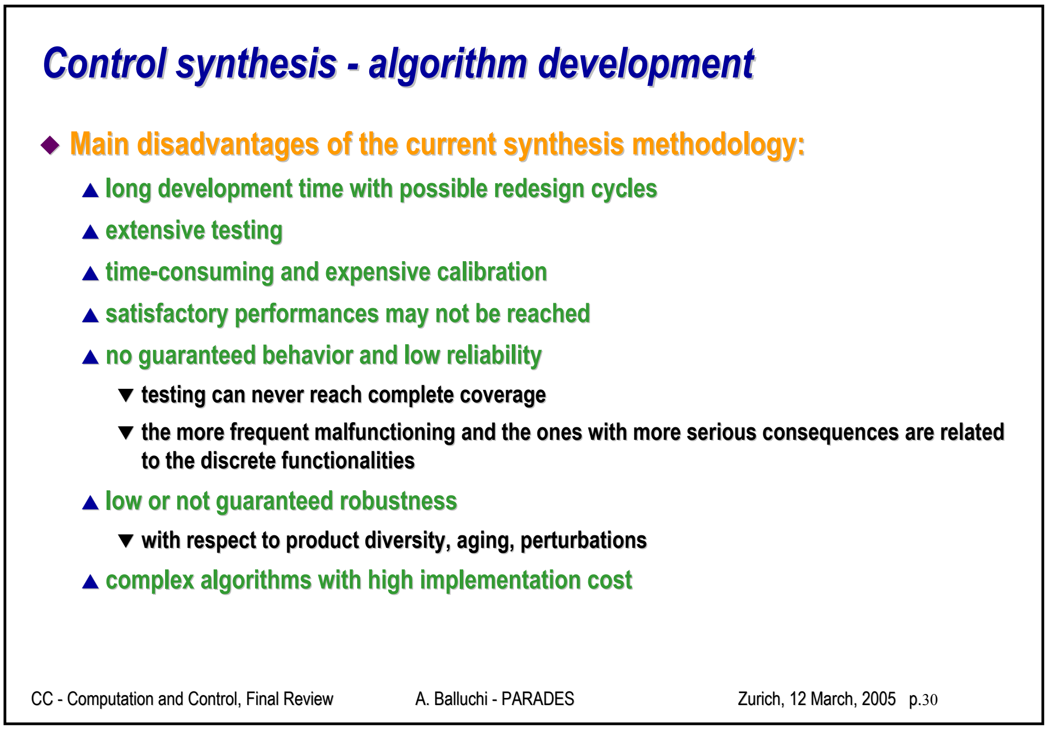
Control synthesis - algorithm development

- Characteristics of the overall electronic control system
  - Multi-rate control system composed of nested control loops that interact with other embedded controllers
    - frequency and phase drifts between sampling frequencies
    - event driven actions
    - asynchronous communication on the network
  - Implements both continuous and discrete functionalities
    - more discrete than continuous
    - control algorithms may have many operation modes
      - nominal operation modes
      - safety, protection and recovery modes
    - computations performed at transition time are very important
      - switching conditions
      - controller initializations
  - Complexity: more than 150 I/O and 200 algorithms in engine control units
  - Many algorithms for diagnosis, fault tolerance and safety (x-by-wire systems)



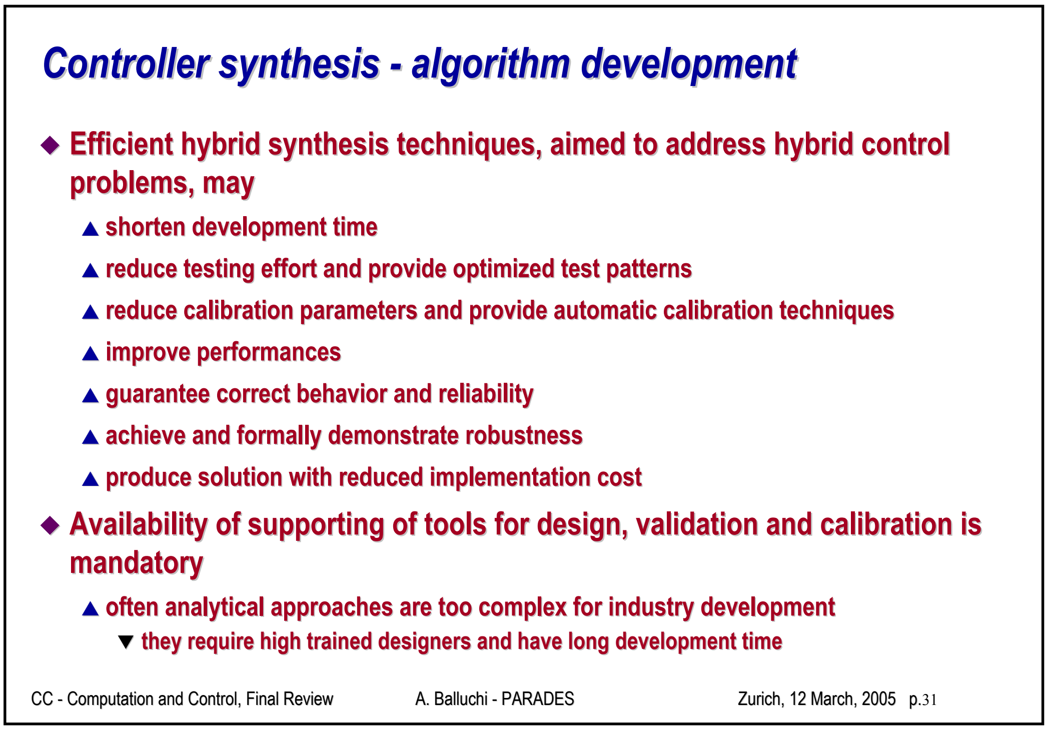
Control synthesis - algorithm development

- If the algorithm can be obtained by derivation, then
  - Only minor changes are realized on an existing algorithm to meet specification
  - The resulting algorithm is often not optimized (excess of discrete modes)
- Otherwise, a new algorithm is developed
  - Desired specification and accurate plant model are often hybrid
  - Current methodology
    - continuous functionalities (in each operation mode) design
      - based on mean-value models with ad hoc solutions for hybrid issues and critical behaviors
      - observer-based and internal-model like schemes are used to cope with nonlinearities
    - discrete functionalities design
      - direct implementation from specifications (no analytical techniques as in hw design)
    - no structured approach to integrated hybrid design
    - first, the algorithm is designed for nominal operating conditions, then critical maneuvers and uncertainties are taken into account



Control synthesis - algorithm development

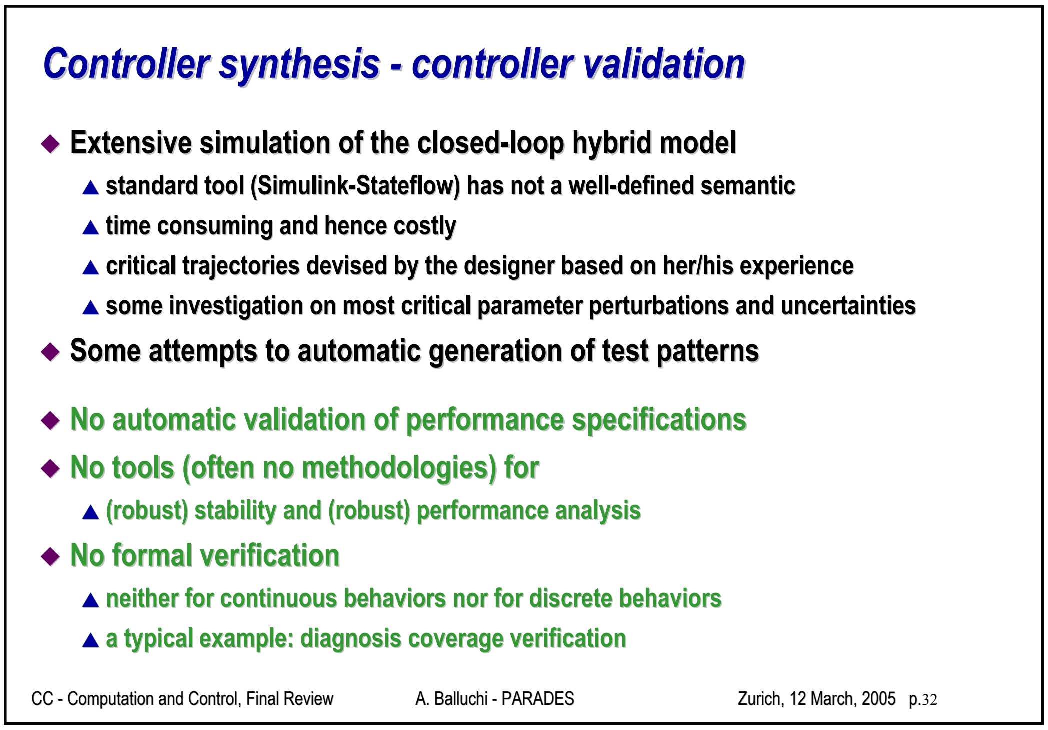
- Main disadvantages of the current synthesis methodology:
  - long development time with possible redesign cycles
  - extensive testing
  - time-consuming and expensive calibration
  - satisfactory performances may not be reached
  - no guaranteed behavior and low reliability
    - testing can never reach complete coverage
    - the more frequent malfunctioning and the ones with more serious consequences are related to the discrete functionalities
  - low or not guaranteed robustness
    - with respect to product diversity, aging, perturbations
  - complex algorithms with high implementation cost



Controller synthesis - algorithm development

- Efficient hybrid synthesis techniques, aimed to address hybrid control problems, may
  - shorten development time
  - reduce testing effort and provide optimized test patterns
  - reduce calibration parameters and provide automatic calibration techniques
  - improve performances
  - guarantee correct behavior and reliability
  - achieve and formally demonstrate robustness
  - produce solution with reduced implementation cost
- Availability of supporting of tools for design, validation and calibration is mandatory
  - often analytical approaches are too complex for industry development
    - they require high trained designers and have long development time

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.31



Controller synthesis - controller validation

- Extensive simulation of the closed-loop hybrid model
  - standard tool (Simulink-Stateflow) has not a well-defined semantic
  - time consuming and hence costly
  - critical trajectories devised by the designer based on her/his experience
  - some investigation on most critical parameter perturbations and uncertainties
- Some attempts to automatic generation of test patterns
- No automatic validation of performance specifications
- No tools (often no methodologies) for
  - (robust) stability and (robust) performance analysis
- No formal verification
  - neither for continuous behaviors nor for discrete behaviors
  - a typical example: diagnosis coverage verification

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.32

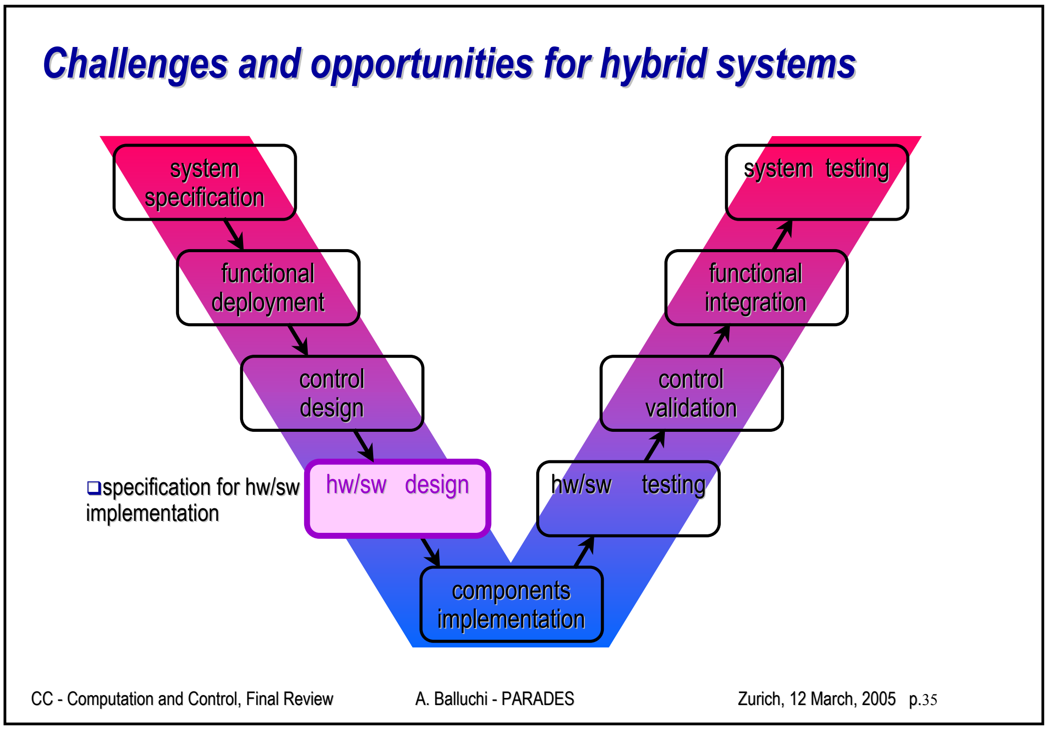
Controller synthesis - controller validation

- Tools for efficient simulation of closed-loop hybrid models
- Hybrid system validation and verification have to be automatic: methodologies and tools should be developed to address in industrial size problems
  - (robust) stability and (robust) performance analysis
  - (robust) invariant set computation
  - automatic validation of performance specification
  - automatic test pattern generation
- Computation of conservative approximations for the largest sets of
  - plant parameter uncertainties
  - calibration parameters
  - implementation parameters (e.g. sampling-time and latency)

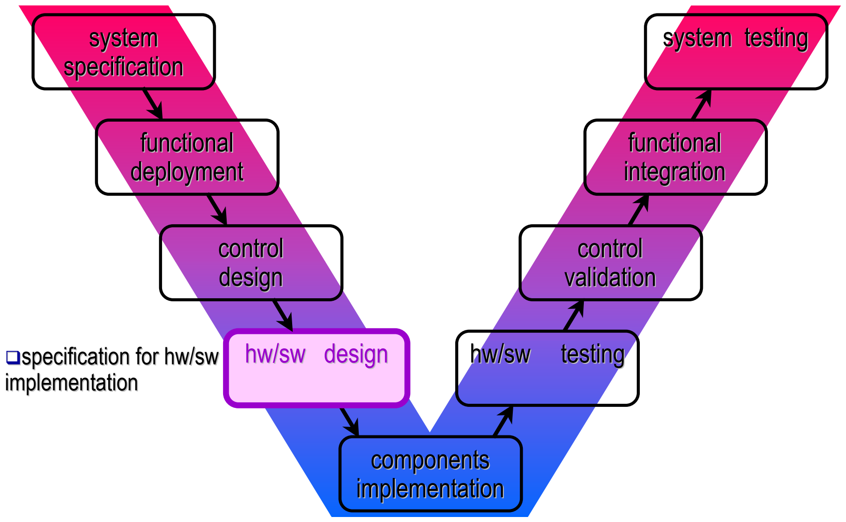
  for which the desired performances are achieved

Control design

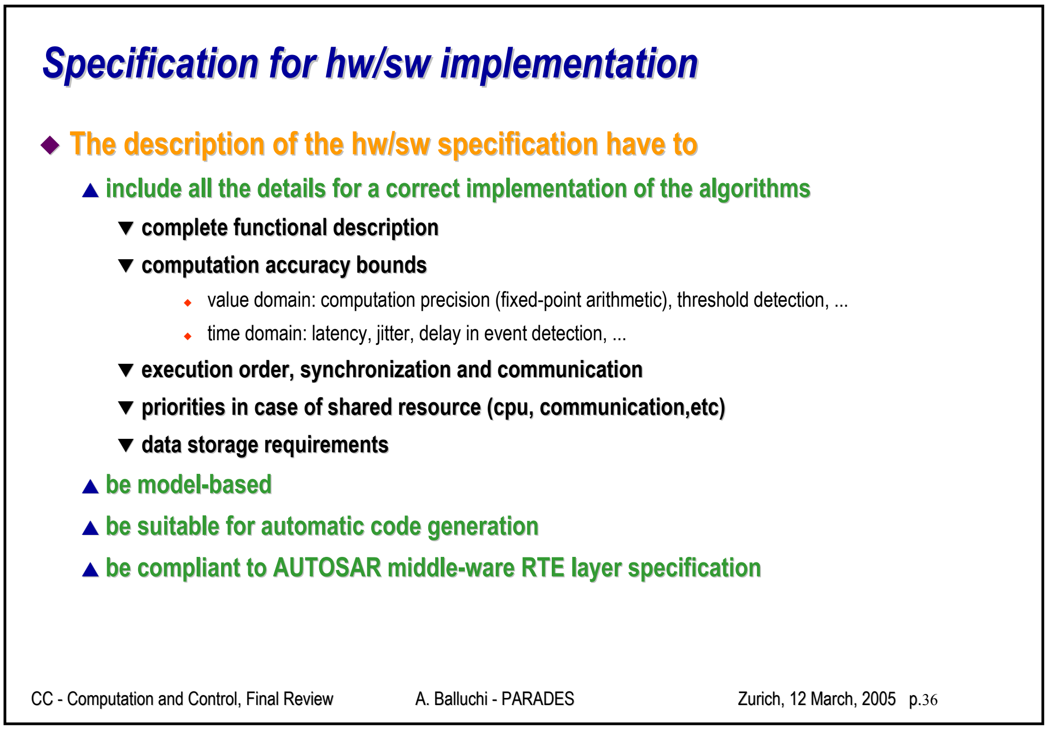
- Plant modeling
  - model development
  - identification
  - validation
- Controller synthesis
  - plant model and specifications analysis
  - algorithms development
  - controller validation
- **Fast prototyping**



Challenges and opportunities for hybrid systems



CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.35



Specification for hw/sw implementation

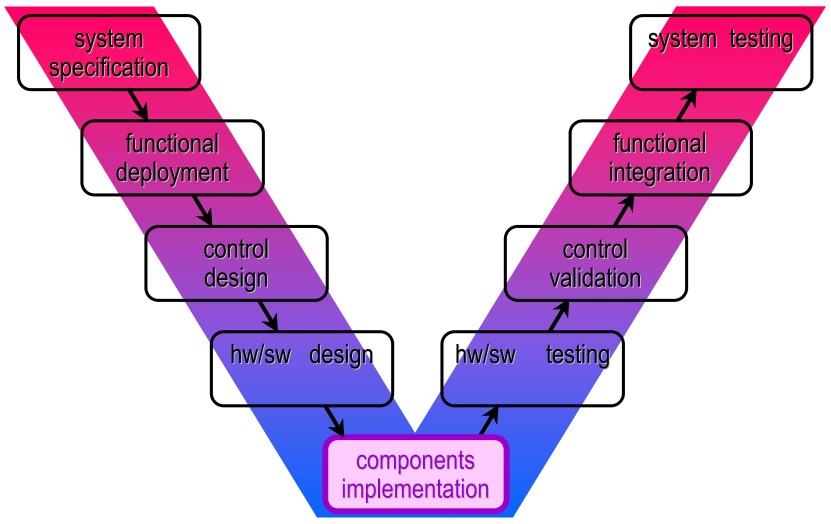
- The description of the hw/sw specification have to
  - include all the details for a correct implementation of the algorithms
    - complete functional description
    - computation accuracy bounds
      - value domain: computation precision (fixed-point arithmetic), threshold detection, ...
      - time domain: latency, jitter, delay in event detection, ...
    - execution order, synchronization and communication
    - priorities in case of shared resource (cpu, communication,etc)
    - data storage requirements
  - be model-based
  - be suitable for automatic code generation
  - be compliant to AUTOSAR middle-ware RTE layer specification

CC - Computation and Control, Final Review A. Balluchi - PARADES Zurich, 12 March, 2005 p.36

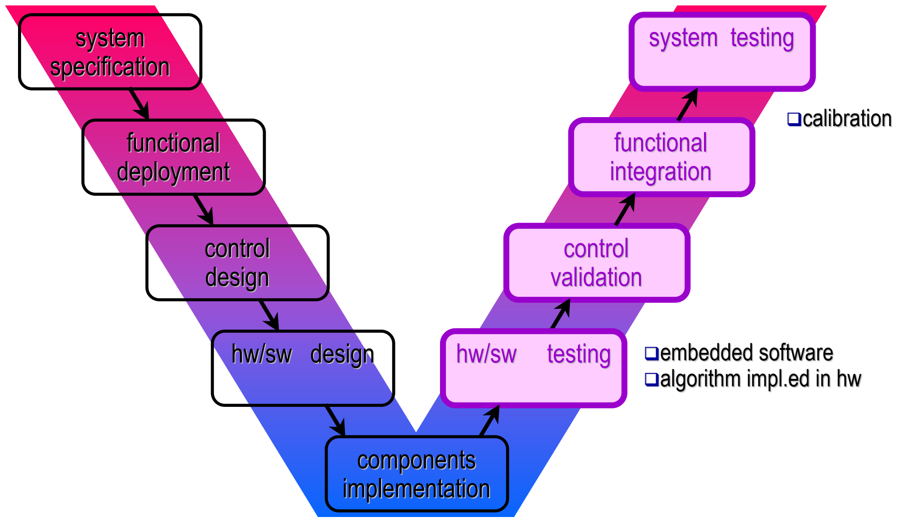
Specification for hw/sw implementation

- Hybrid formalisms can successfully support the description of the implementation specification
- Methodologies and tools for the definition and validation of implementation constraints should be developed
  - the degradation produced by the implementation of control algorithms on bounded resource hardware has to be exported and modeled at the control system layer to obtain acceptance criteria for the hw/sw implementation
- Tools supporting the specification for hw/sw implementation have to
  - allow the description of the implementation constraints and acceptance criteria
  - be efficiently integrated with software development tools
  - either provide automatic code generation or be linked to auto-coding tools

Challenges and opportunities for hybrid systems

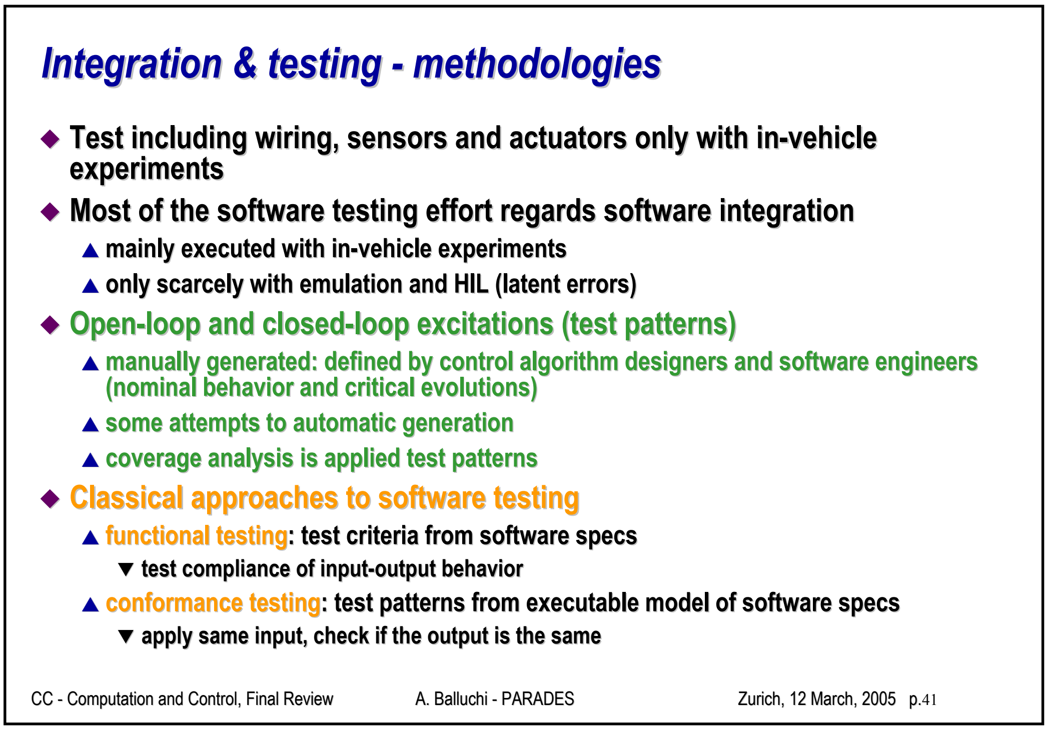


Challenges and opportunities for hybrid systems



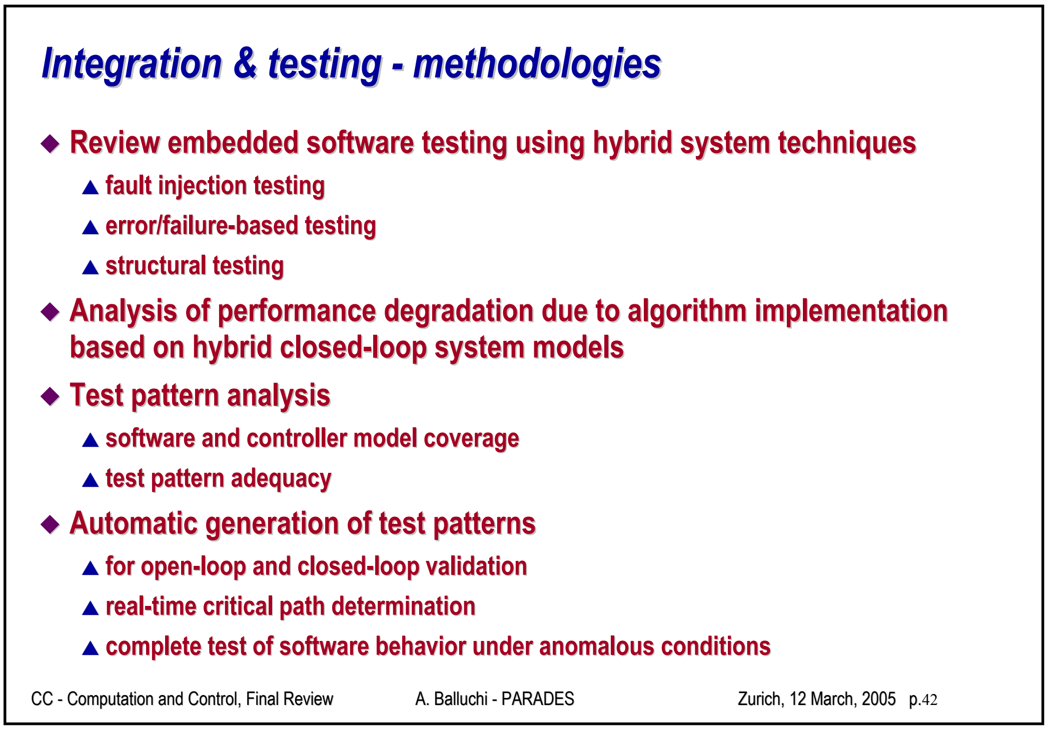
Integration & testing - methodologies

- microcontroller emulation
  - objective: ECU test (including power electronics) step by step
  - type: open-loop test with manual input excitation, HIL or in-vehicle
  - check: by-inspection check of compliance with software specification
- software in the loop (SIL)
  - objective: source code functional test (no real-time)
  - type: closed-loop simulation of compiled software and plant model
  - check: compliance with closed-loop specification and closed-loop simulation of controller model
- processor in the loop (PIL)
  - objective: target processor executable code functional test (no real-time)
  - type: closed-loop simulation of executable code and plant model
  - check: compliance with closed-loop specification and closed-loop simulation of controller model
- hardware in the loop (HIL)
  - objective: real-time ECU test (including power electronics but not wiring)
  - type: black-box test of ECU in closed-loop with real-time virtual plant
  - check: compliance with closed-loop specification
- additional testing for software quality, communication over the network, etc
- in-vehicle experiments (fast prototyping and diagnosis tools)



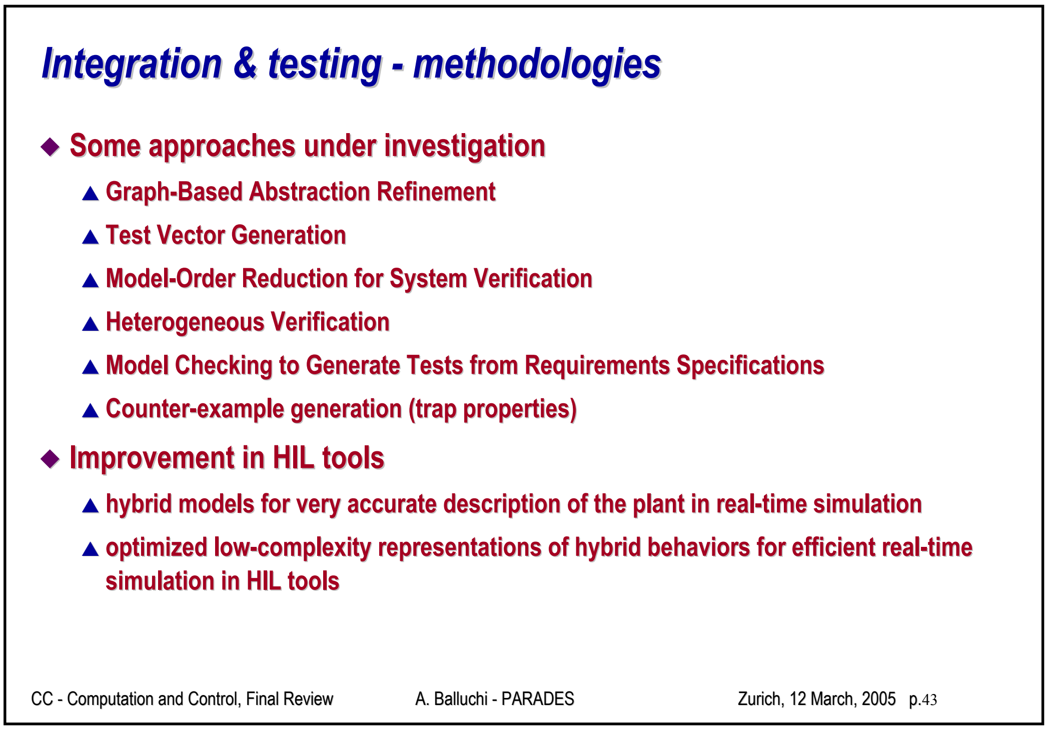
Integration & testing - methodologies

- Test including wiring, sensors and actuators only with in-vehicle experiments
- Most of the software testing effort regards software integration
  - mainly executed with in-vehicle experiments
  - only scarcely with emulation and HIL (latent errors)
- Open-loop and closed-loop excitations (test patterns)
  - manually generated: defined by control algorithm designers and software engineers (nominal behavior and critical evolutions)
  - some attempts to automatic generation
  - coverage analysis is applied test patterns
- Classical approaches to software testing
  - functional testing: test criteria from software specs
    - test compliance of input-output behavior
  - conformance testing: test patterns from executable model of software specs
    - apply same input, check if the output is the same



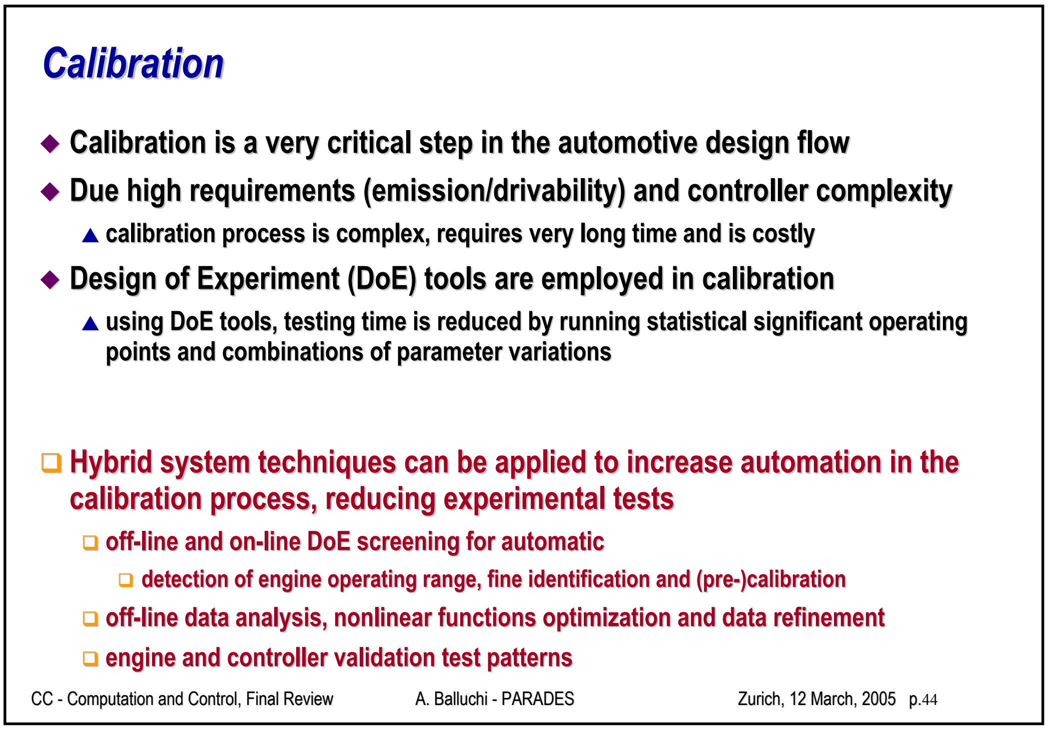
Integration & testing - methodologies

- Review embedded software testing using hybrid system techniques
  - fault injection testing
  - error/failure-based testing
  - structural testing
- Analysis of performance degradation due to algorithm implementation based on hybrid closed-loop system models
- Test pattern analysis
  - software and controller model coverage
  - test pattern adequacy
- Automatic generation of test patterns
  - for open-loop and closed-loop validation
  - real-time critical path determination
  - complete test of software behavior under anomalous conditions



Integration & testing - methodologies

- Some approaches under investigation
  - Graph-Based Abstraction Refinement
  - Test Vector Generation
  - Model-Order Reduction for System Verification
  - Heterogeneous Verification
  - Model Checking to Generate Tests from Requirements Specifications
  - Counter-example generation (trap properties)
- Improvement in HIL tools
  - hybrid models for very accurate description of the plant in real-time simulation
  - optimized low-complexity representations of hybrid behaviors for efficient real-time simulation in HIL tools



Calibration

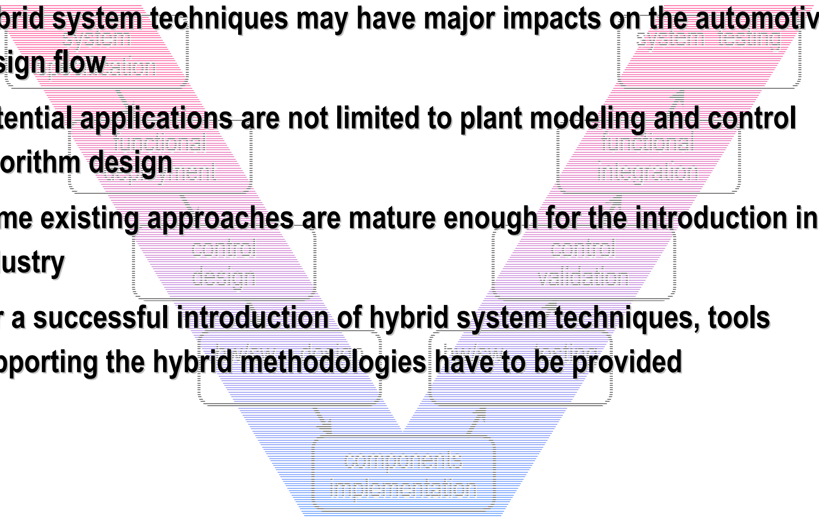
- Calibration is a very critical step in the automotive design flow
- Due high requirements (emission/drivability) and controller complexity
  - calibration process is complex, requires very long time and is costly
- Design of Experiment (DoE) tools are employed in calibration
  - using DoE tools, testing time is reduced by running statistical significant operating points and combinations of parameter variations

- Hybrid system techniques can be applied to increase automation in the calibration process, reducing experimental tests
  - off-line and on-line DoE screening for automatic
    - detection of engine operating range, fine identification and (pre-)calibration
  - off-line data analysis, nonlinear functions optimization and data refinement
  - engine and controller validation test patterns

Acknowledgements

Alberto Ferrari, Pierpaolo Murrieri
PARADES, Rome, I
Luca Benvenuti
Univ. of Rome, I
Gabriele Serra, Giacomo Gentile,Carlo Siviero, Walter Nesci
Magneti Marelli Powertrain, Bologna, I
Paolo Ferracin
CNH, Modena, I
Stefan Kowalewski
RWTH, Aachen, G (formerly at Bosch)
Gilberto Burgio
Ford, Aachen, G
Pandeli Borodani
Centro Ricerche Fiat, Torino, I

Conclusions



- **Hybrid system techniques may have major impacts on the automotive design flow**
- **Potential applications are not limited to plant modeling and control algorithm design**
- **Some existing approaches are mature enough for the introduction in the industry**
- **For a successful introduction of hybrid system techniques, tools supporting the hybrid methodologies have to be provided**