Space Missions Model-Based Control vs. Intelligent Control

Dr.-Ing. Johann Bals
Institute of System Dynamics and Control
DLR - German Aerospace Center
Oberpfaffenhofen, Germany

1st International Round Table on Intelligent Control for Space Missions

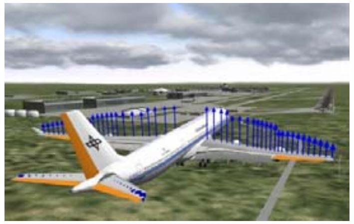
ESTEC , Noordwijk, NL
24 Nov 2017

DLR System Dynamics and Control

Fields of Application

Space

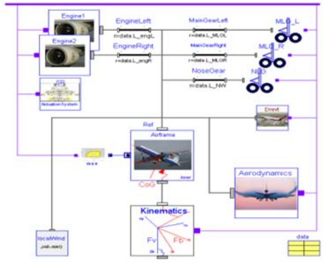
Aeronautics



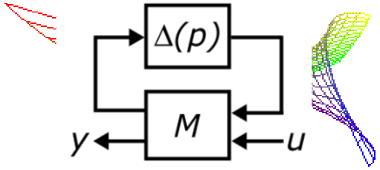
Ground Vehicles

Tech-Transfer

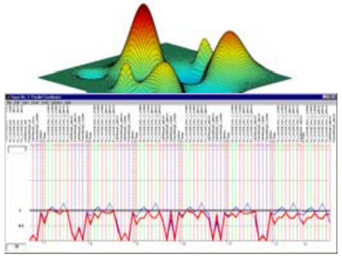
Modeling and Simulation



Control



Optimisation



Methods and Tools, Design Process

Model-Based Control vs. Intelligent Control

- **Model based control synthesis:** Models are used for synthesis of control laws
    - LQG, Hinfinity, …
- **Embedded model control:** Model of the system to be controlled is directly incorporated into the feedforward or feedback controller
    - Model Predictive Control
    - Dynamic Inversion / Feedback Linearization
    - Inverse Model Feedforward Control

Intelligent Control:

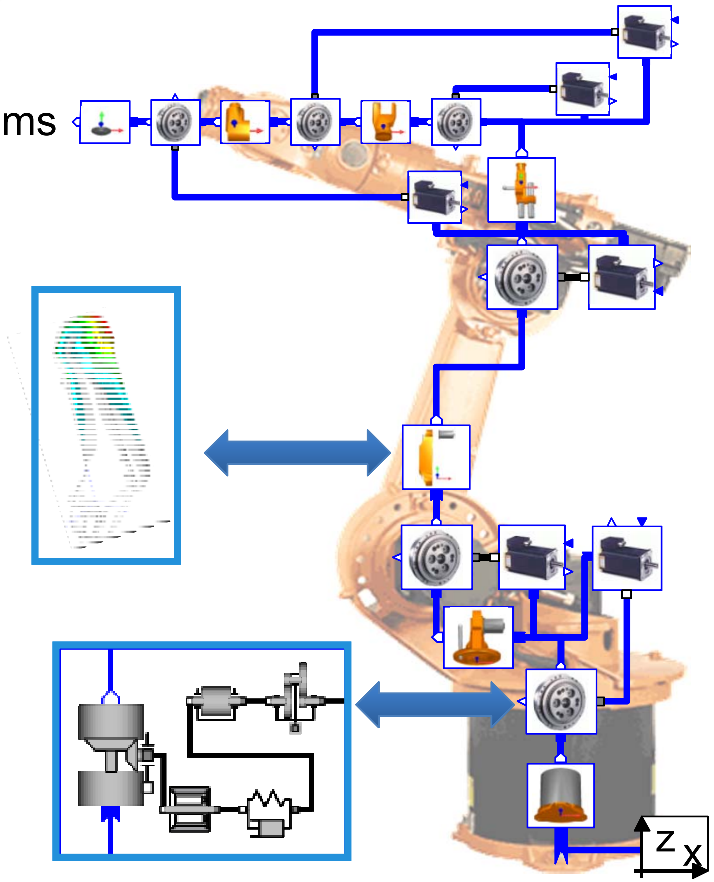
- Neural Networks control
- Bayesian control
- Fuzzy Logic control
- Expert Systems and Artificial Intelligence
- Genetic and Evolutionary control

Model based control design process:

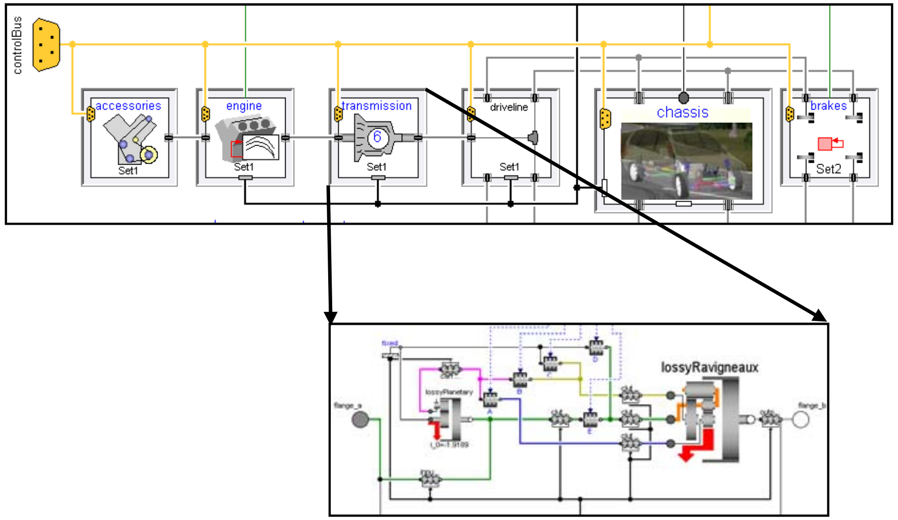
- Use of modeling and simulation for performance assessment, robustness analysis and parameter tuning of control algorithms
- Automatic code generation and qualification

System Dynamics Modelling with Modelica

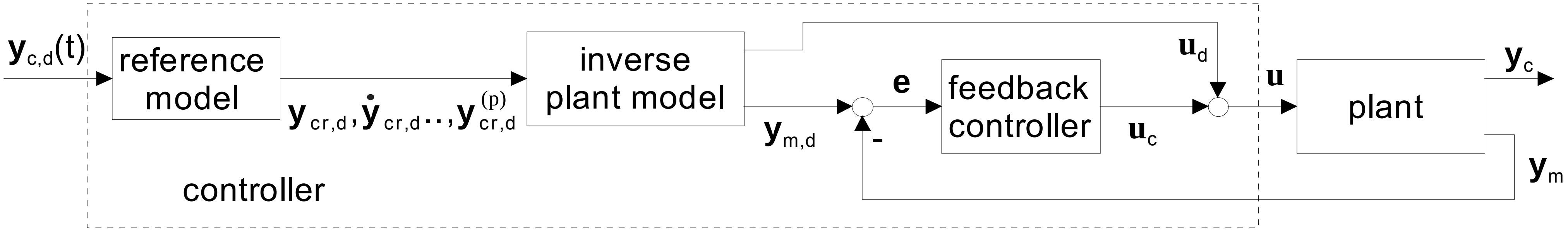
- Physical modelling and simulation of complex multi-domain systems
- For design, optimisation, control, verification, virtual testing
- Open standard developed by the Modelica Association
- Chair RMC-SR



Example: Power Train Library



Inverse Models with Modelica/Dymola



Inverse plant model computes desired actuator and desired measurement signals based on non-linear plant model.

Generating inverse models with Modelica

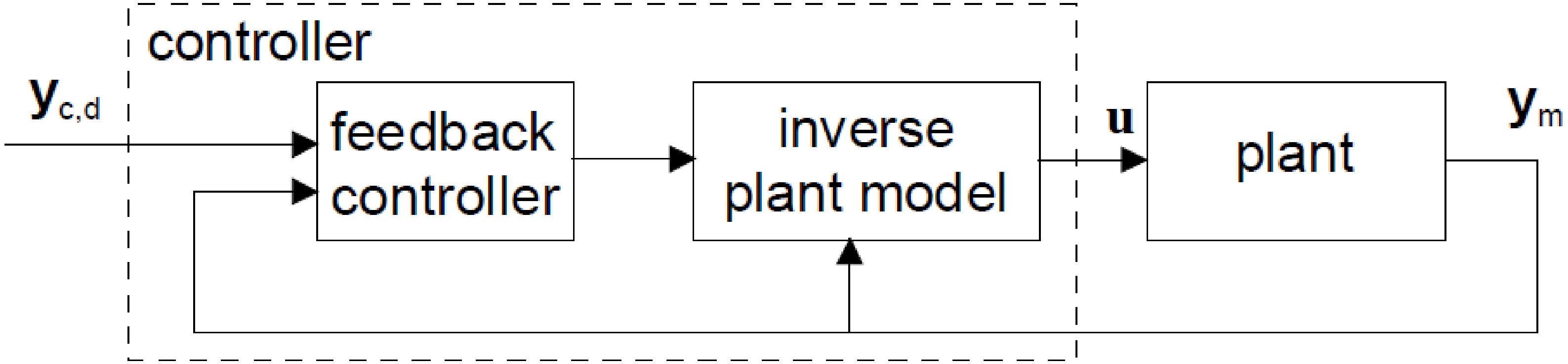
- An inverse model of the DAE is constructed by exchanging the meaning of variables:
- A subset of the input vector u, is treated no longer as known but as unknown,

and previously unknown variables from the vectors x and/or y are treated as known inputs.

Inverse models are also DAEs in the form $\mathbf{0} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, \mathbf{u})$

- The result is still a DAE which can be handled with the same methods
- Allows to generate two degree of freedom controller for non-linear plant model

Inverse Models with Modelica/Dymola



Generating inverse models with Modelica

- An inverse model of the DAE is constructed by exchanging the meaning of variables:
- A subset of the input vector u, is treated no longer as known but as unknown,

and previously unknown variables from the vectors x and/or y are treated as known inputs.

Inverse models are also DAEs in the form $\mathbf{0} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, \mathbf{u})$

- The result is still a DAE which can be handled with the same methods
- Allows to generate two degree of freedom controller for non-linear plant model

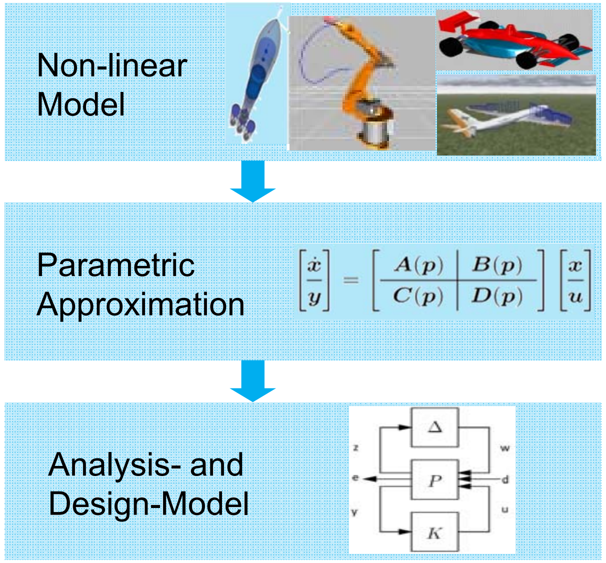
Advanced Methods and Tools for Robust Control Design and Analysis

Challenges

- Robust stability for time varying parameters
- Low complexity uncertainty modelling

Research Topics

- Efficient toolchains
- Application for highly non-linear and uncertain systems

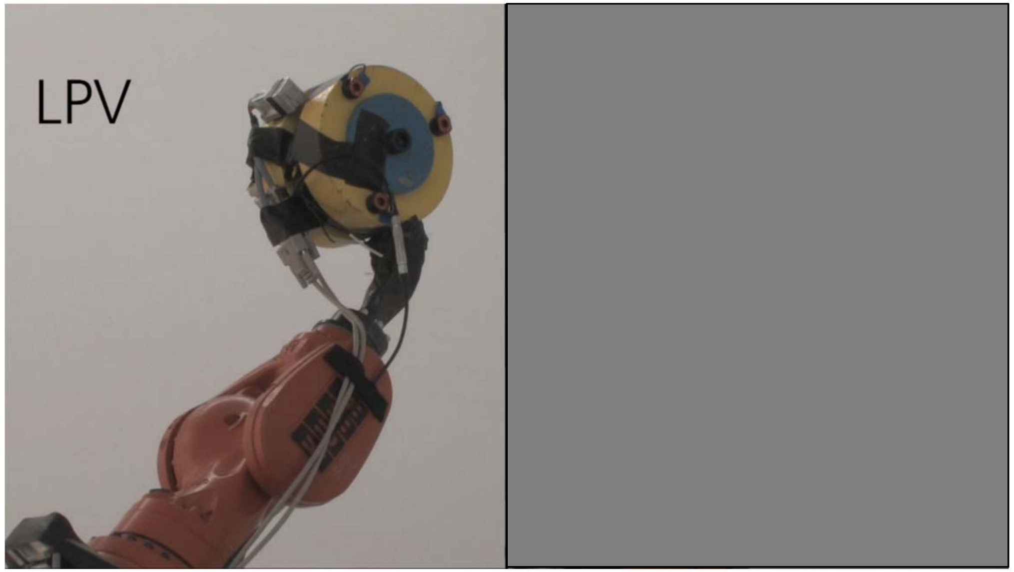


Non-linear Model

Parametric Approximation

$$\begin{bmatrix}\dot{x}\\ y\end{bmatrix} = \left[\begin{array}{c|c} A(p) & B(p)\\ \hline C(p) & D(p)\end{array}\right]\begin{bmatrix}x\\ u\end{bmatrix}$$

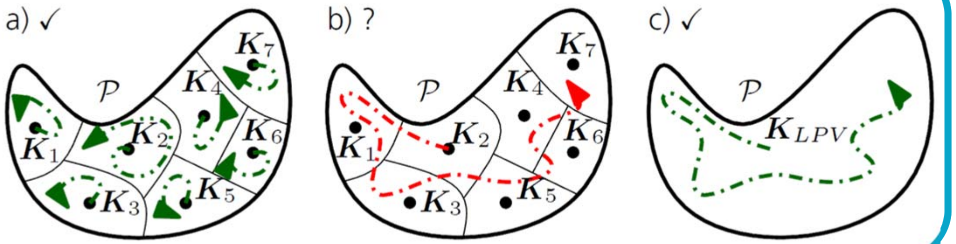
Analysis- and Design-Model



a) Common design approach with gain scheduling
→ Local stability only

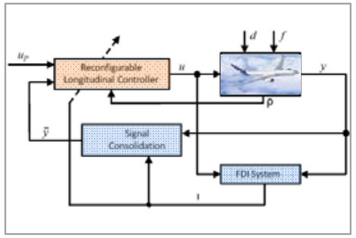
b) Interpolation between linear time-varying controllers
→ Scheduling between controller is critical

c) Linear Parameter-Varying control (LPV)
→ Guaranteed stability for the whole envelope



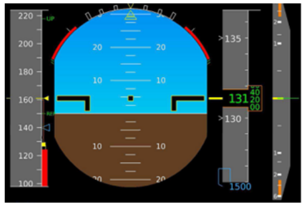
Further Core Areas in Model-Based Control

- Robust Fault-detection and -Isolation
  - Actuators
  - Sensors
  - Combining signal and model based methods



Controller Reconfiguration based on FDI information

- Fault Tolerant Control
  - Prediction and flight envelope protections
  - Robust control

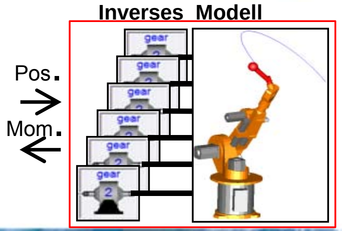


Indication of safe flight envelope in the Primary Flight Display

- Health Monitoring

Early Detection of System Degradation

- Inverse Models for Path-Planning

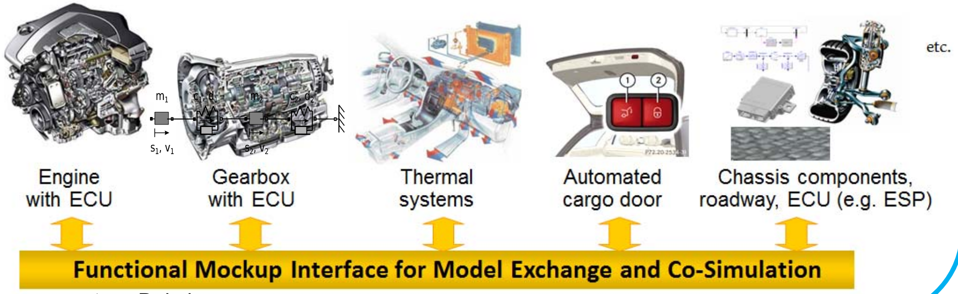


Automatic generation of inverse models:
Input: Positions
Output: Forces and Moments

European Research Projects (ITEA/BMBF)

MODELISAR 2008 – 2011, 27 Mill. Euro

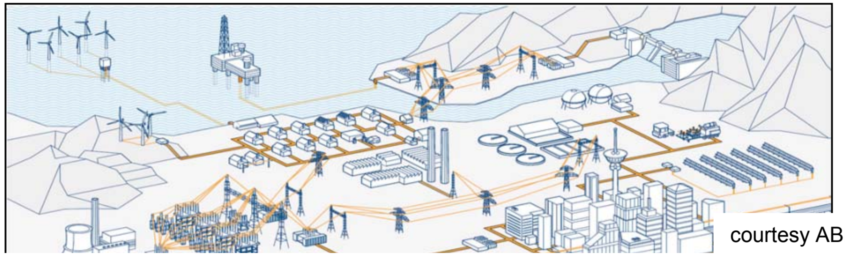
Organized by Daimler, DLR, and others.
FMI-Standard for model exchange and co-simulation
In 2017: supported by > 100 tools



courtesy Daimler

MODRIO 2012 – 2016, 22 Mill. Euro

Organized by EDF, DLR and others.
Nonlinear models for requirements and online operations



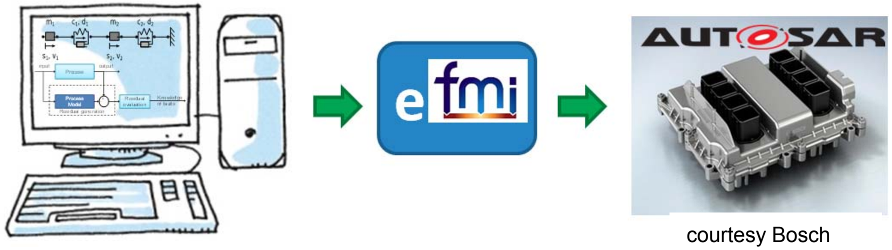
courtesy ABB

In 2017 by ABB: > 7.5 % of Germanies power production (5000 MW) are generated with Modelica/FMI based online optimization

EMPHYSIS 2017 – 2020, 15 Mill. Euro

Organized by Bosch, DLR, and others.
Nonlinear Models in ECU production code
(FMI for embedded systems)

- OEMs + Tier1 suppliers (use cases)
- Vendors of simulation tools (Dymola, SimulationX, ...)
- Vendors of ECU tools (TargetLink, ASCET, ...)
- Research institutes (DLR, ...)



courtesy Bosch

Optimization as Design Tool

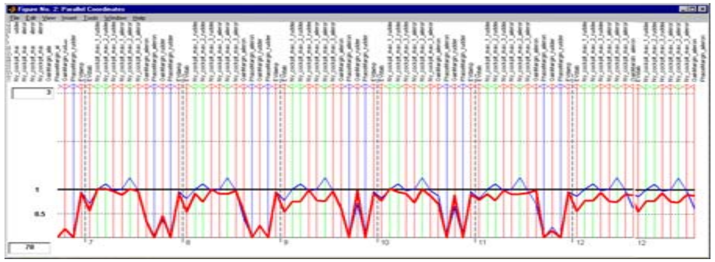
Applications:

- Robust Control Design
- Parameter ID
- System Design
- Design Verification
- Optimal feedforward contr.
- Real-Time Optimization

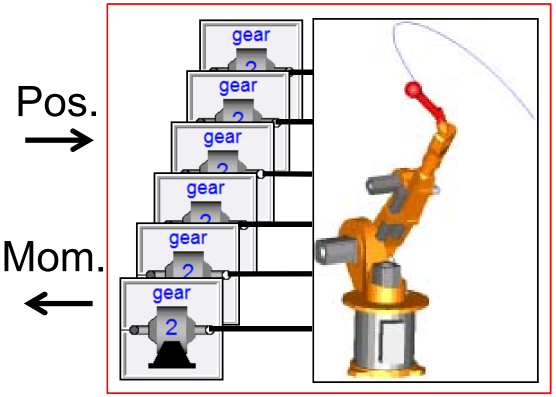
Techniques:

- Multi-objective-Optimization
- Multi-Case-Optimization
- Worst-Case-Optimization
- Pareto-Front Search
- Optimal Path-Planning
- using inverse models

Multi-Objective Optimization: Criteria in parallel Coordinates



Inverse Model



Tools:

- Matlab: MOPS
- Multi-Phase Optimal Path Planning: trajOpt
- For Modelica: Optimization Library in Catia/Dymola

Specialties:

- Advanced Numeric Algorithms
- Multi-Shooting Algorithms
- FMI-Connection
- Monte-Carlo-Analysis
- Parallel Computing
- User-friendly

Optimization as Design Tool

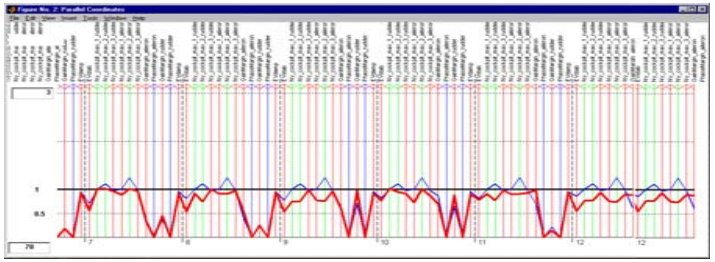
Applications:

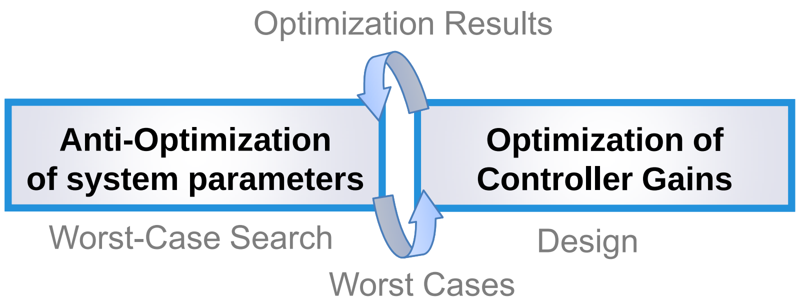
- Robust Control Design
- Parameter ID
- System Design
- Design Verification
- Optimal feedforward contr.
- Real-Time Optimization

Techniques:

- Multi-objective-Optimization
- Multi-Case-Optimization
- Worst-Case-Optimization
- Pareto-Front Search
- Optimal Path-Planning
- using inverse models

Multi-Objective Optimization: Criteria in parallel Coordinates





Optimization Results

Anti-Optimization of system parameters

Optimization of Controller Gains

Worst-Case Search

Design

Worst Cases

Tools:

- Matlab: MOPS
- Multi-Phase Optimal Path Planning: trajOpt
- For Modelica: Optimization Library in Catia/Dymola

Specialties:

- Advanced Numeric Algorithms
- Multi-Shooting Algorithms
- FMI-Connection
- Monte-Carlo-Analysis
- Parallel Computing
- User-friendly

Model-Based Control Design Process

System-Requirements

Approval

Specification and Design

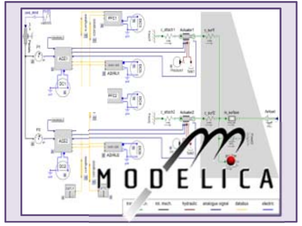
Implementation

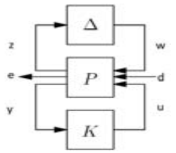
Integration And Test

Requirement Analysis

- *Requirements Library*
- *Funktion-model*

System Architecture



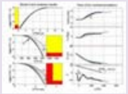


Submodule Design

MOPS Entwurfsoptimierung

S/W H/W Implementation

Submodule Tests

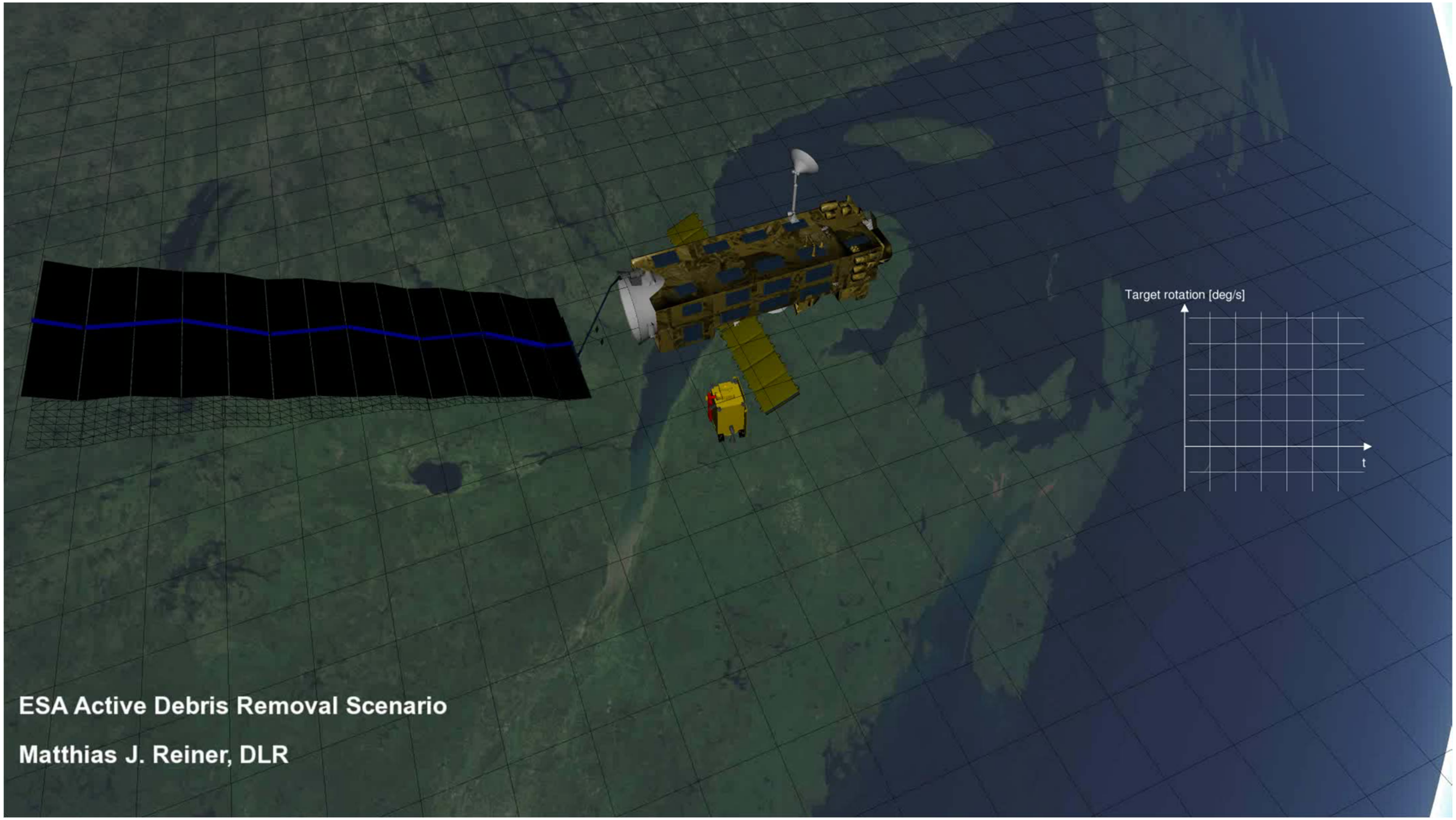


MOPS Assessment:
- *Worst-Case Search*
- *Monte-Carlo Analysis*

Integration

Validation

Parameter-ID

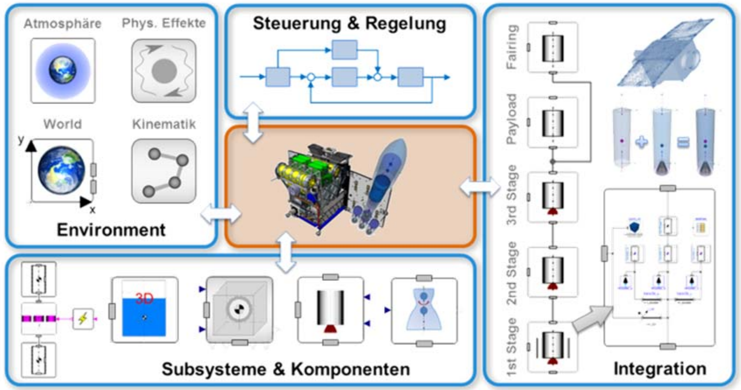


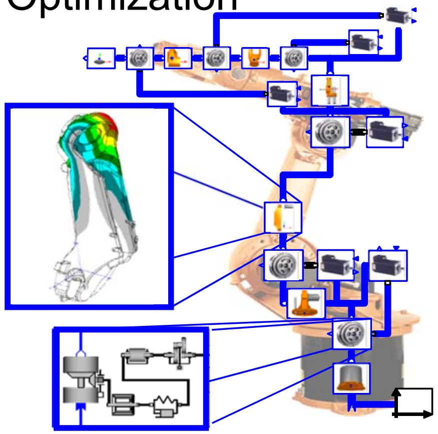
ESA Active Debris Removal Scenario

Matthias J. Reiner, DLR

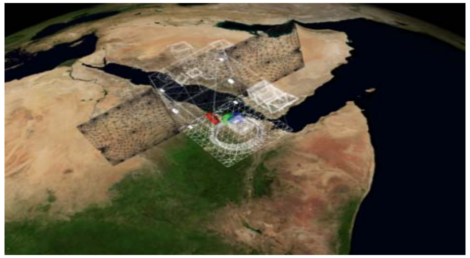
Object-oriented Modeling with Modelica

- Modelica SpaceSystems & Environment Library
  - Modeling of environment and orbit
  - Actuator and sensor models
- Modelica Robots & RobotDynamics Library
  - Robot models
  - Kinematic & Dynamic
- Supporting Modelica Libraries
  - Multi-body, FlexibleBodies, Visualization, Optimization

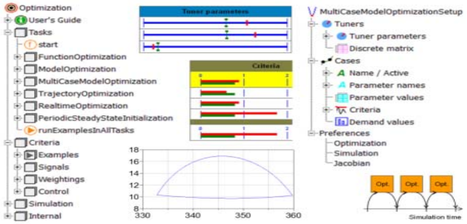




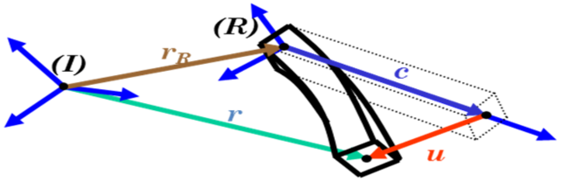
- Visualization



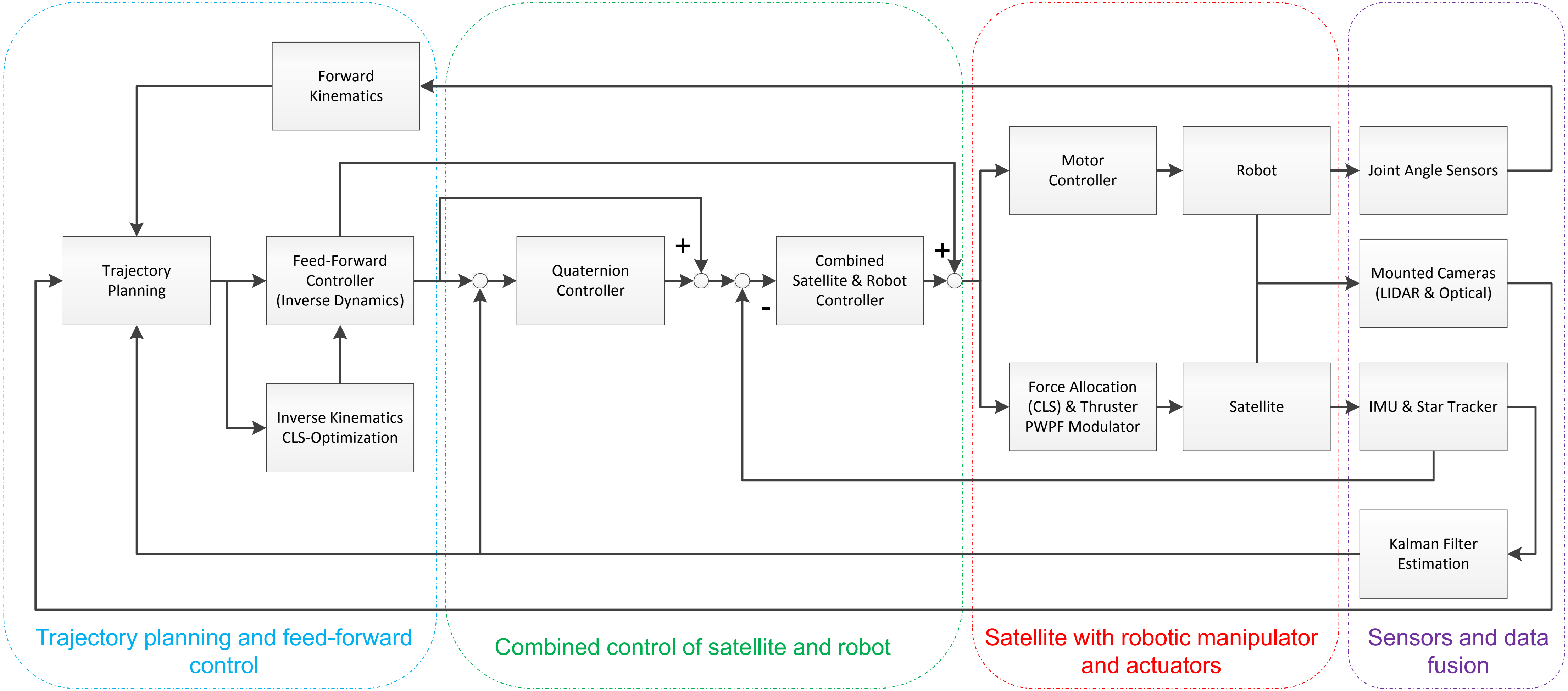
- Optimizaiton



- Flexible bodies

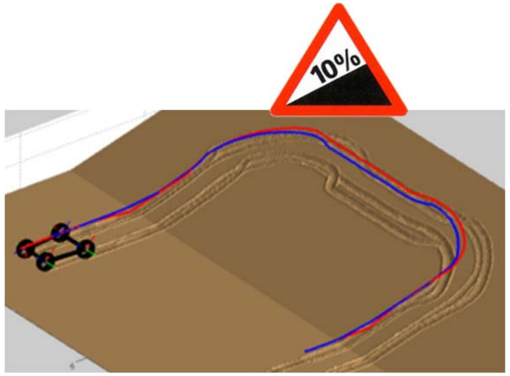


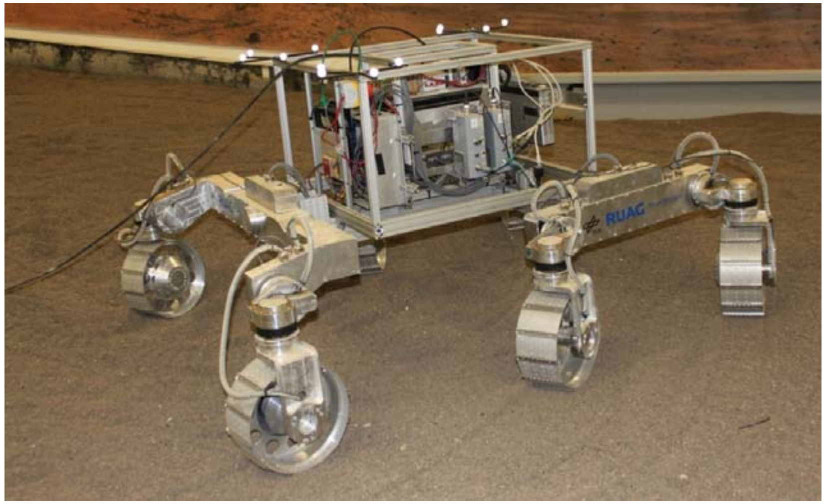
Overview of the combined satellite and robot arm control

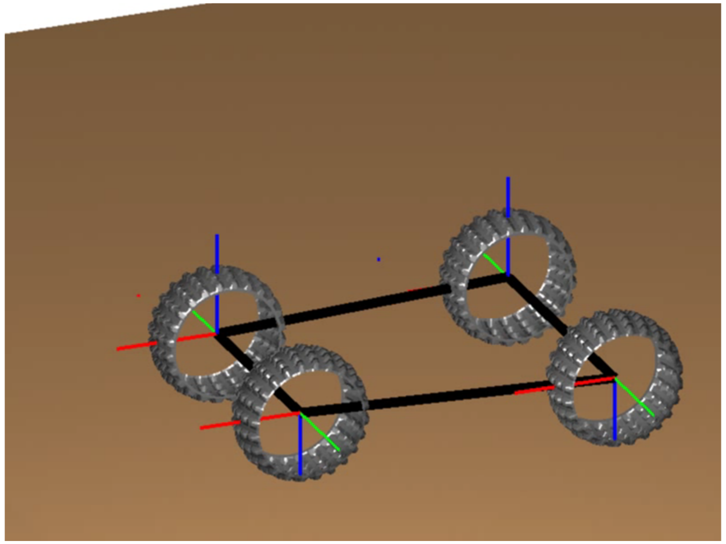


Model-Based Rover Controls
Example: Model-Predictive Control

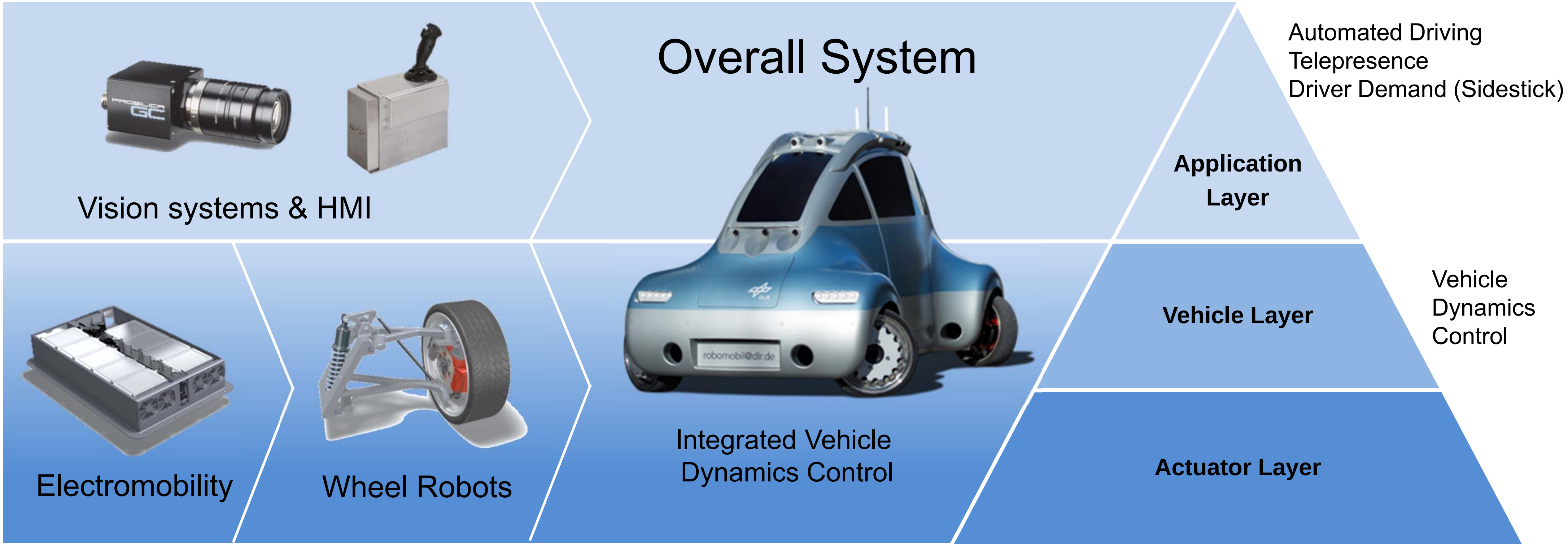
- Goal: Optimal control of the wheels with respect to robust locomotion and energy consumption
- Method: MPC with rover dynamics and simplified terramechanics models
- Evaluation: Within the EGP rover in simulation and Exomars BB2 in the DLR planetary exploration lab







Control Architecture of ROboMObil



Vision systems & HMI

Electromobility

Wheel Robots

Overall System

Integrated Vehicle Dynamics Control

Application Layer

Automated Driving
Telepresence
Driver Demand (Sidestick)

Vehicle Layer

Vehicle
Dynamics
Control

Actuator Layer

Vehicle Level

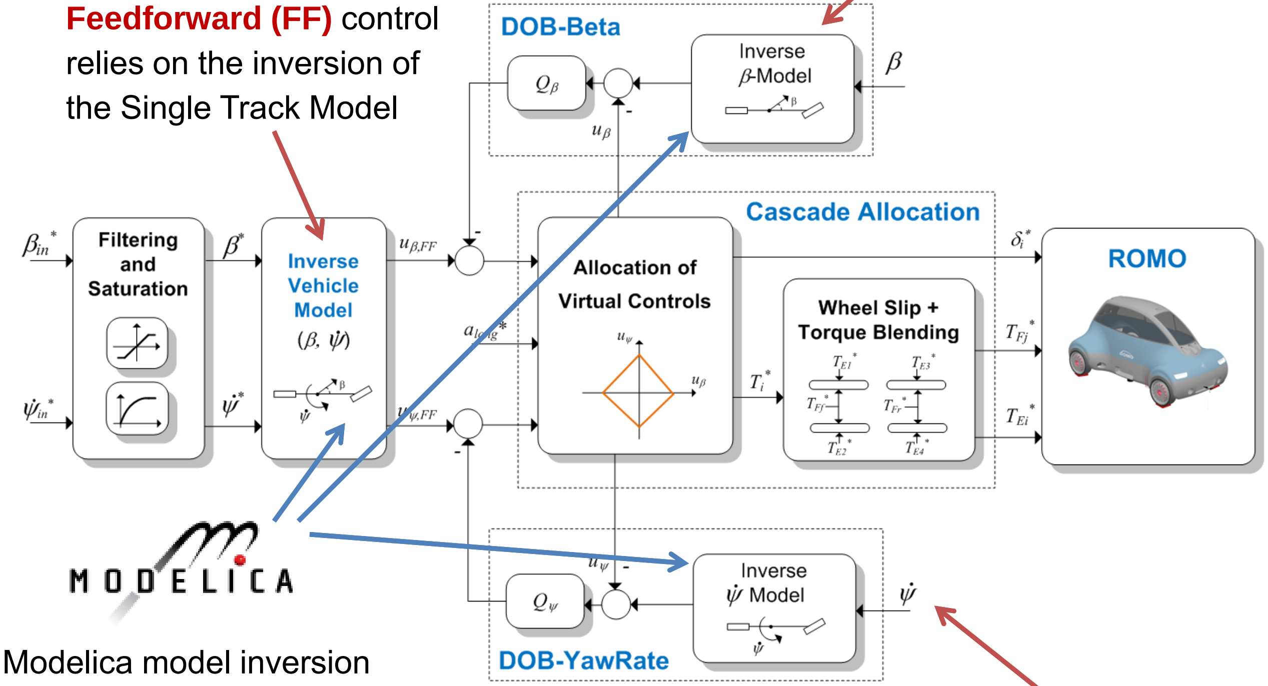
Extended Vehicle Dynamics Control (VDC 3.x) - 1

- Execution of motion demands in hierarchical controller structure

- **Exploitation of the full potential of the wheel robot dynamics bandwidth for active driving stabilization through optimization based methods (2-DoF Q-Loop Control & Control Allocation)**
- Optimization of energy recuperation also during wheel slip control and in demanding handling situations through robust MPC approaches (Guaranteed reachability of the control variable trough truncation of a control variable reserve)

Feedback (FB) control of the vehicle‘s yaw-rate and side-slip based on the disturbance observer (DOB) principle

Feedforward (FF) control relies on the inversion of the Single Track Model



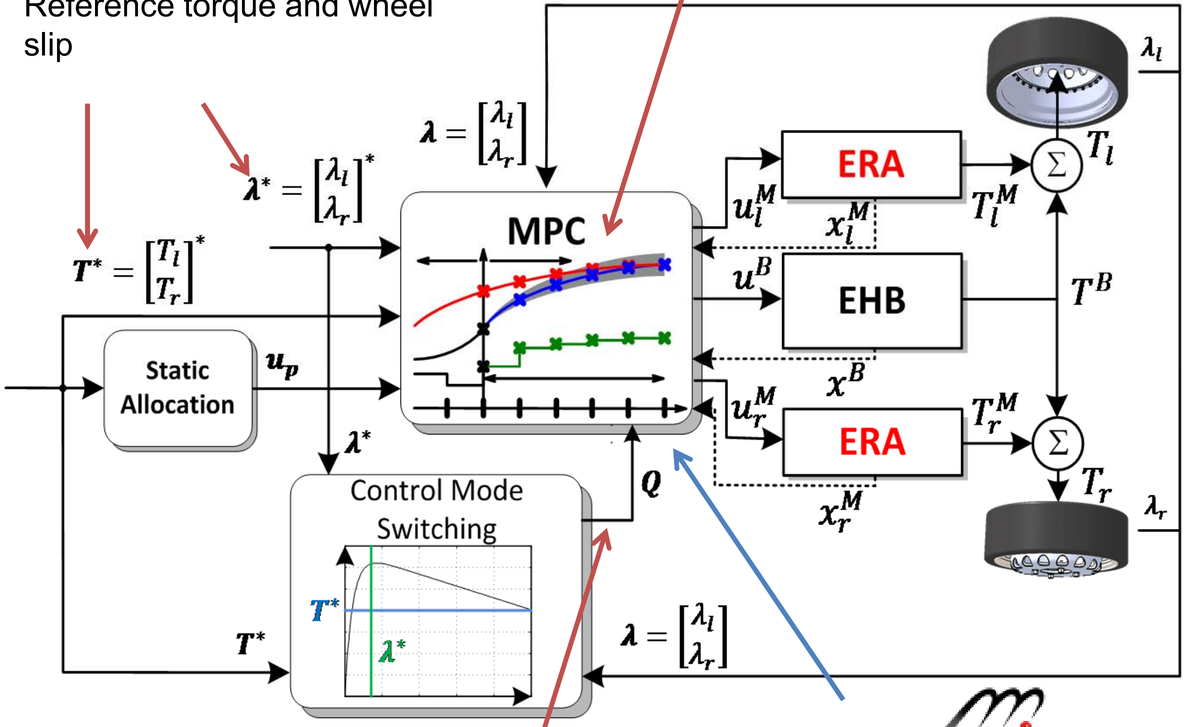
Modelica model inversion

To facilitate the distribution of the actuation effort, a two-step **cascade allocation** process, was developed.

Vehicle Level

Extended Vehicle Dynamics Control (VDC 3.x) - 2

- Execution of motion demands in hierarchical controller structure
- Exploitation of the full potential of the wheel robot dynamics bandwidth for active driving stabilization through optimization based methods (2-DoF Q-Loop Control & Control Allocation)
- **Optimization of energy recuperation also during wheel slip control and in demanding handling situations through robust MPC approaches (Guaranteed reachability of the control variable trough truncation of a control variable reserve)**



Minimization of the cost function with consideration of system dynamics

Reference torque and wheel slip

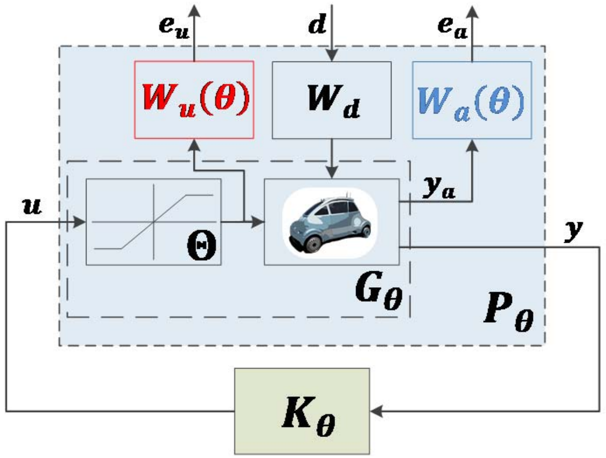
Control mode switching through weighting

DLR Modelica toolbox for robust MPC

Actuator Level

Vertical Dynamics

- Manipulation of the wheel loads and body movement for vehicles like ROMO with high unsprung masses
- Robust LPV control methods with high bandwidth and inverse semi-active damper model



Application Level
Reactive Obstacle Avoidance

- Calculation based on two consecutive images from a monocular camera.
- Detection of collisions with static and dynamic obstacles.
- Direct motion correction derived from velocities in the image space without intermediate transformation in Cartesian coordinates

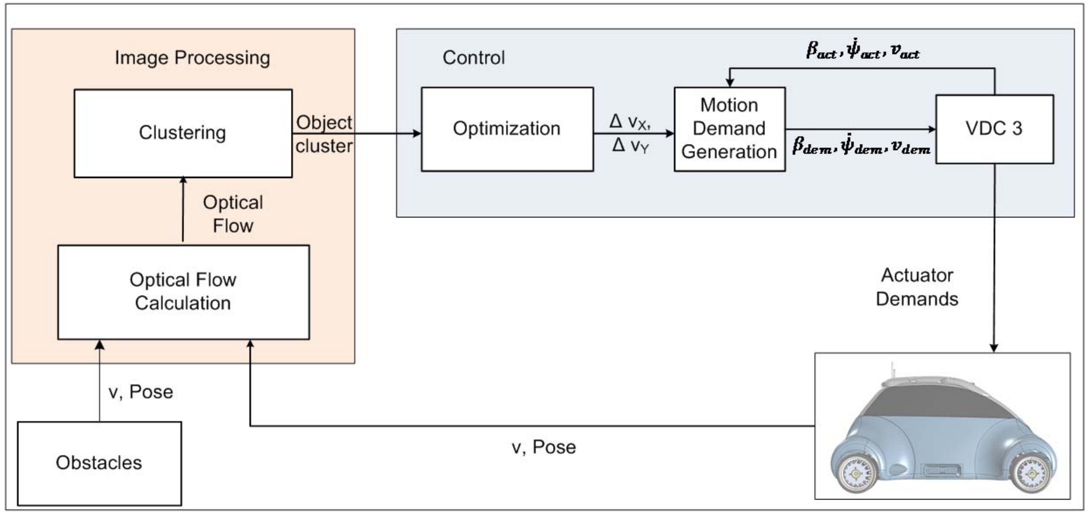


Image Processing

Clustering

Object cluster

Optical Flow

Optical Flow Calculation

v, Pose

Obstacles

Control

Optimization

Δ v$_X$, Δ v$_Y$

Motion Demand Generation

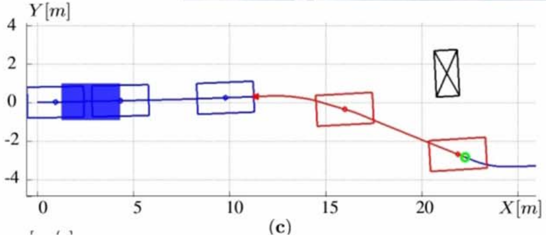
$\beta_{act}, \dot{\psi}_{act}, v_{act}$

$\beta_{dem}, \dot{\psi}_{dem}, v_{dem}$

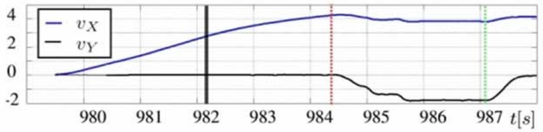
VDC 3

Actuator Demands

v, Pose



(c)



GNC: Model-Based vs. Intelligent Control

Guidance

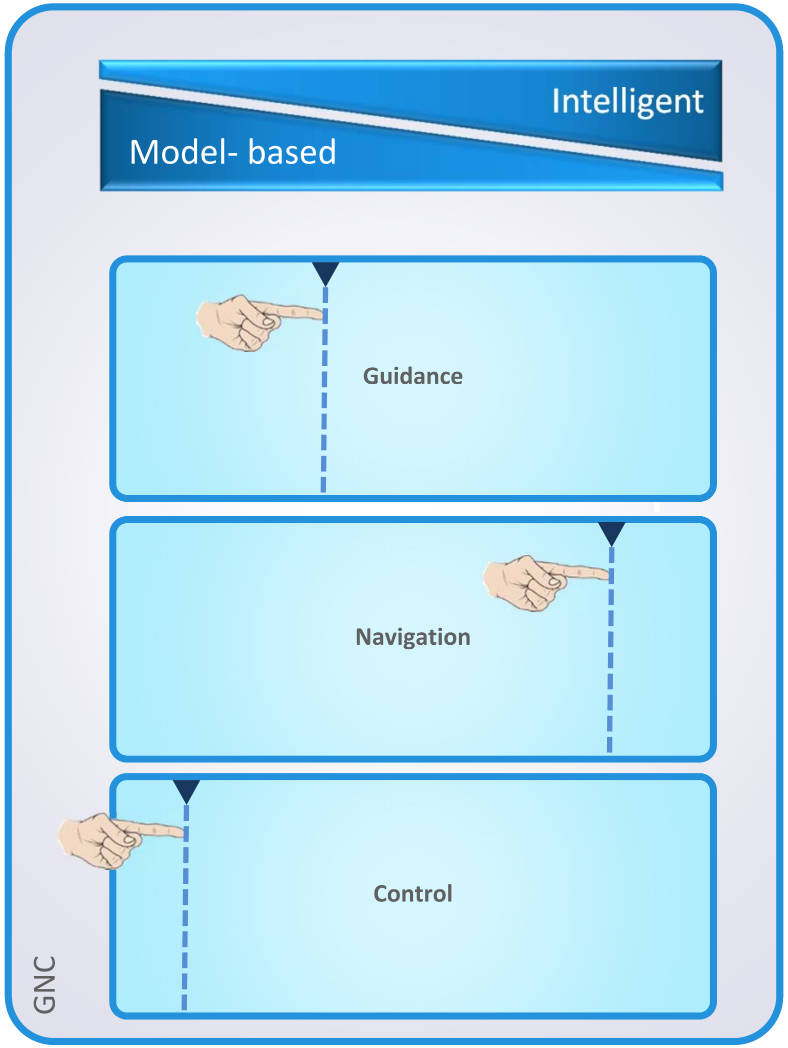
- Optimal guidance and optimization
    - Direct and Indirect methods
- *guidance and motion plans with intelligent control?*
    - Example: soft landing on the moon

Navigation

- Potential for intelligent control in the sense of
    - Vision-based navigation
    - …

Control

- Model-Based control
- First principles => Equations of motion => Physics
- Feedforward (model inversion, approx. inverse models)
- Feedback (LPV, robust control, loop shaping, PID, mu)



Possible round table questions

- ***How could model-based and intelligent control be utilized in a complementary or combined way for future space missions?***

- ***Design process for model-based vs. intelligent control?***
  - ***Certification Requirements***
  - ***Safety Assessments***
  - ***Validation & Verification***