Laboratory CIM & Robotik
Prof. Dipl.-Ing. Georg Stark

Model-based and Component-oriented Programming of Robot Controls

1. Development Process of Industrial Control Units
2. Programming Paradigms
   - object-oriented
   - component-oriented
   - model-based
3. Example – Development of Robot Control ***MRobot***
   - Synchronous Execution and Simulation
   - Sensor Integration

Development Process – Industrial Control Units

Objectives

- Improved Maintainability of Software
- Cost Efficiency
- Optimal Information Flow between Involved Personell
- High Functionality of Software

Approach: Improved Software Technology

Combination of

model-based, component-oriented, object-oriented

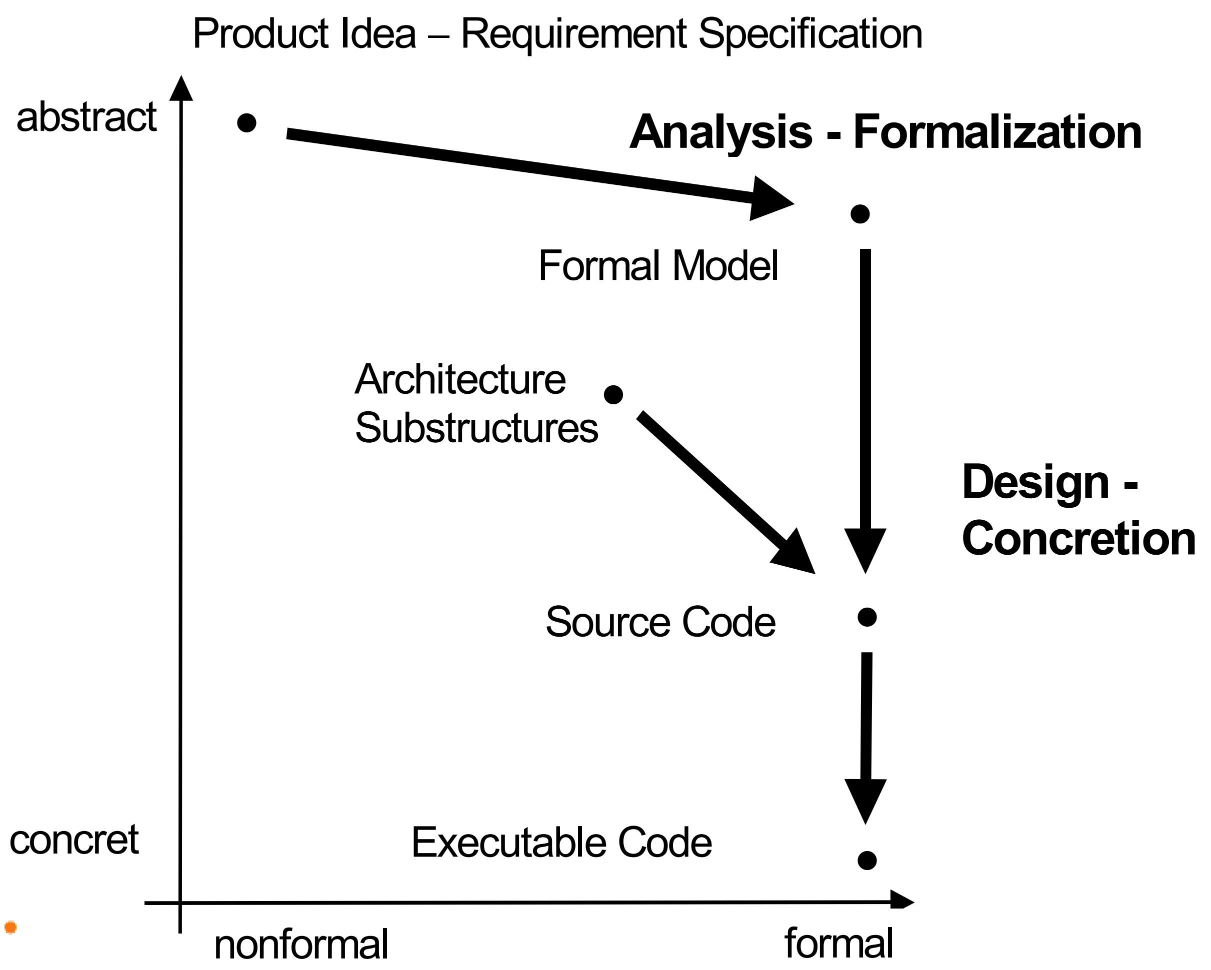
Programming Methods

Development Process - Phases

1. Planning
   - Requirement Specification
2. **Analysis**
   - Knowledge Aquisition,
   - Formal Representation
3. **Design**
   - Architecture
   - Substructures
   - Test Procedure
4. Implementation and Test
5. Verification

Development Process – Analysis and Design

Representation of Information by Using Design Diagram



Object-Oriented Programming

Key Ideas

- Conception of **Classes, Inheritance, Encapsulation**
- Classes are Templates for Software Objects
- Facilitate the Creation of Variants
- **Software Interfaces**, Abstract Classes
- Classes Represent Pieces of Knowledge
- Non-domain-specific Programming Language
- Only Support the Representation of **Structure of Knowledge**, not the Knowledge itself

Component-Oriented Programming

Key Ideas

- Software components are pieces of executable software, to be used via standardized interfaces
- Applying components, a **framework-plugin** architecture can be realized
- The framework supplies the **time-critical functionality**.The **robot-related knowledge** will be implemented by using plugins
- Benefits:
  - Components can be implemented by applying different programming languages
  - The maintenance of software will be improved
- Examples of standardized component interfaces:
  - **COM**, defined by Microsoft,
  - CORBA, OMG (www.omg.org),
  - JavaBeans.

Model-Based Programming

Key Ideas

- Programs implement **knowledge,** related to different domains, e.g. operating dialogues, shape of workpieces, kinematical behavior of machines
- **Formal Models** represent knowledge by applying formal languages
- A model is defined to be a sufficiently precise, coherent representation of a specific area of the real world
- **Domain-specific programming languages** are necessary, to directly and efficiently implement formal models
- The technical software **MATLAB** also comprises a programming language, supporting the direct implementation of formal models

Comparison of Programming Paradigms

- **Object-Oriented Programming**

  Design: Representation of Structure of Knowledge by Classes,
  Safety, Reusability of Software

  Implementation: General Purpose Language

- **Component-Oriented Programming**

  Design: Definition of Executable Structures

  Implementation: Exchangeability (Plugins),
  Various Languages to Be Used

- **Model-Based Programming**

  Analysis: Representation of Knowledge by Formal Models

  Design: Models Define Software Structure

  Implementation: Domain-Specific Language

Example – Robot Control *MRobot*

Functions:

- 1 – 12 Motion Axis
- Interpolation Modes:
  - Point to Point
  - Linear, including Polynom-Bypassing
  - Circular
  - Spline
- **Sensor Control**
- Offline-Programming with
  Realtime Graphical Simulation
- Powerful Application-Specific Programming
  Language (MATLAB Script)

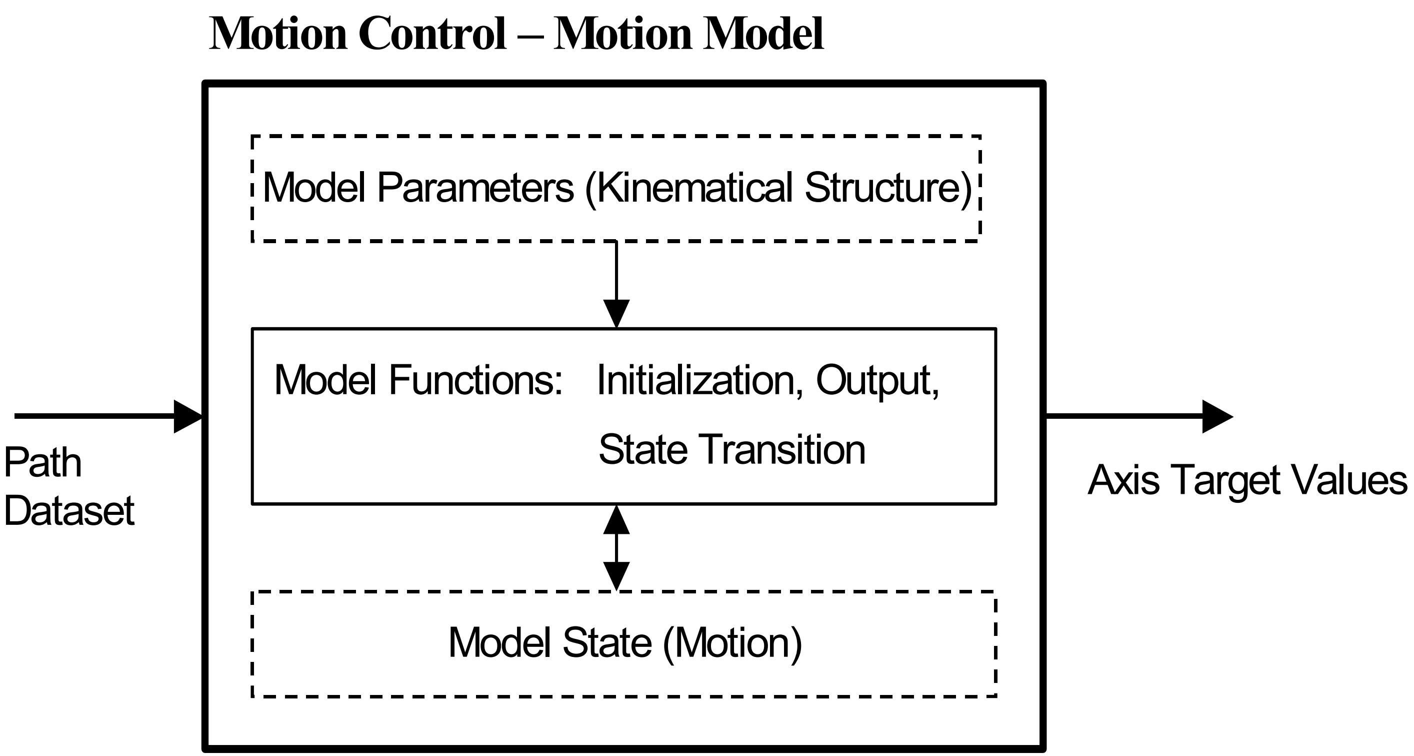
Example – Robot Control *MRobot*

User-Benefits:

- **MATLAB-Interface**:
  - Robot-Systemsoftware
  - Application Software
- **Easy Programming**:
  - applicable by Robot Experts, not having intensive Programming Skills
- **Decreased Costs** for Development and Maintenance

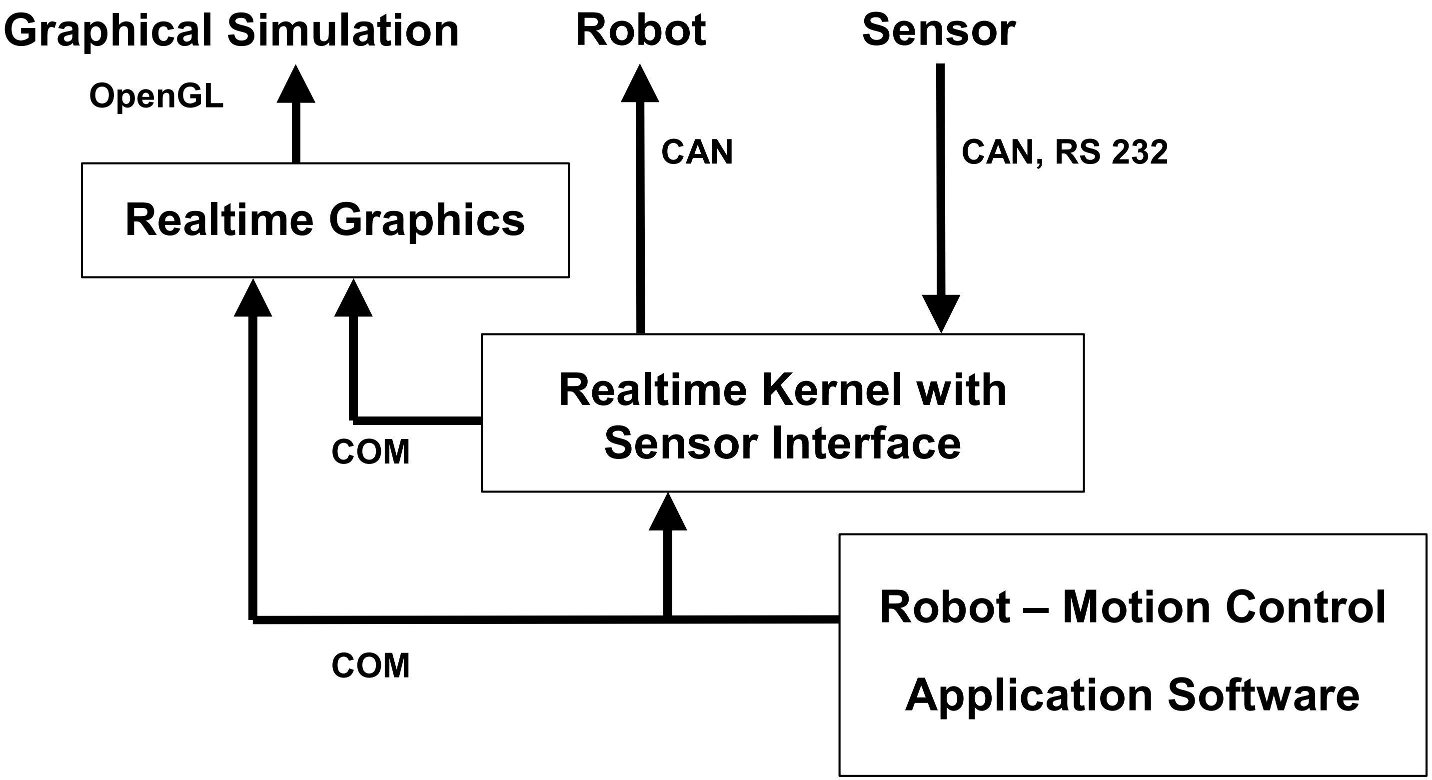
Robot Control *MRobot*

Model-based Design – Motion Control



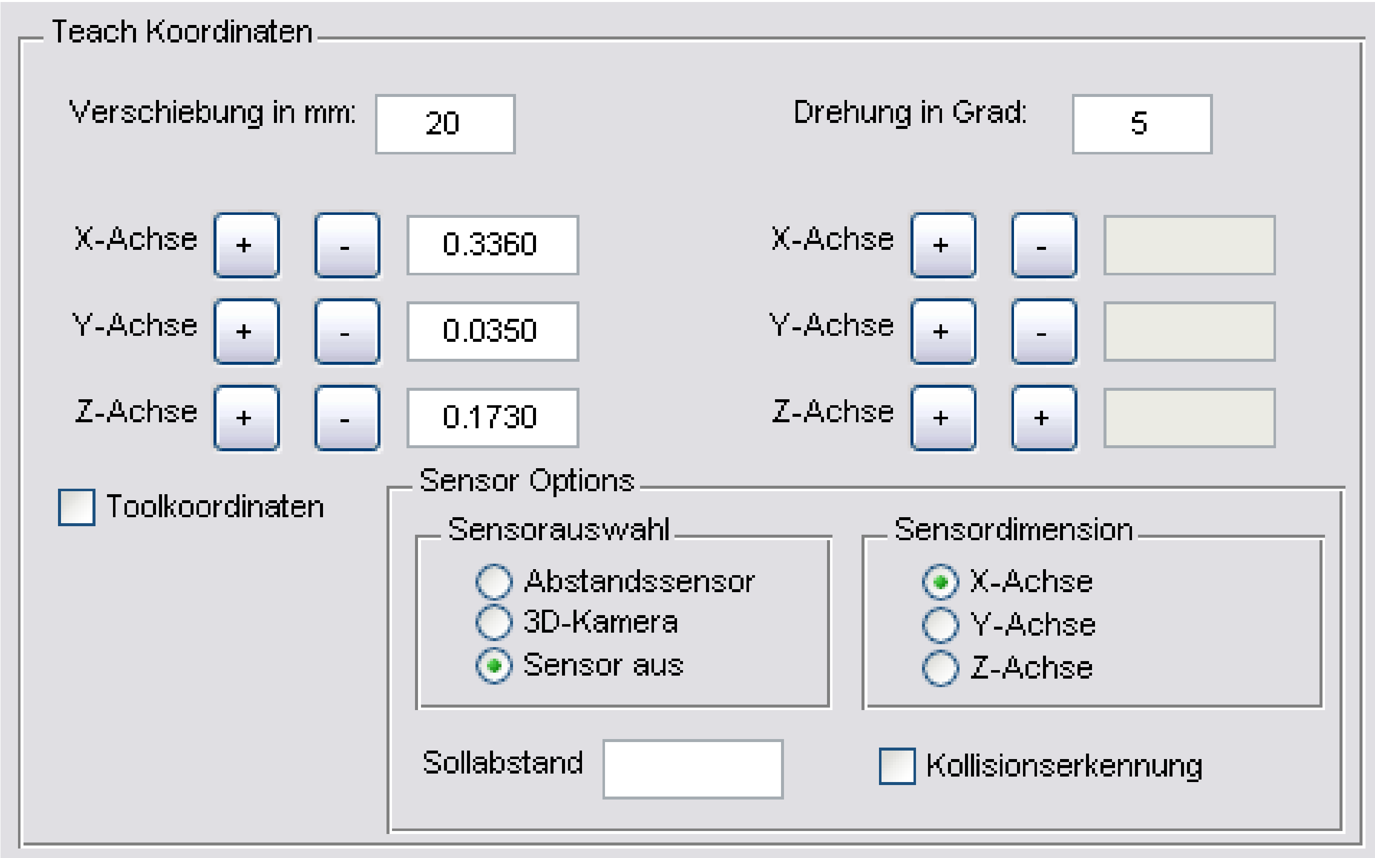
Robot Contol *MRobot*

Software Structure – Components, Interfaces



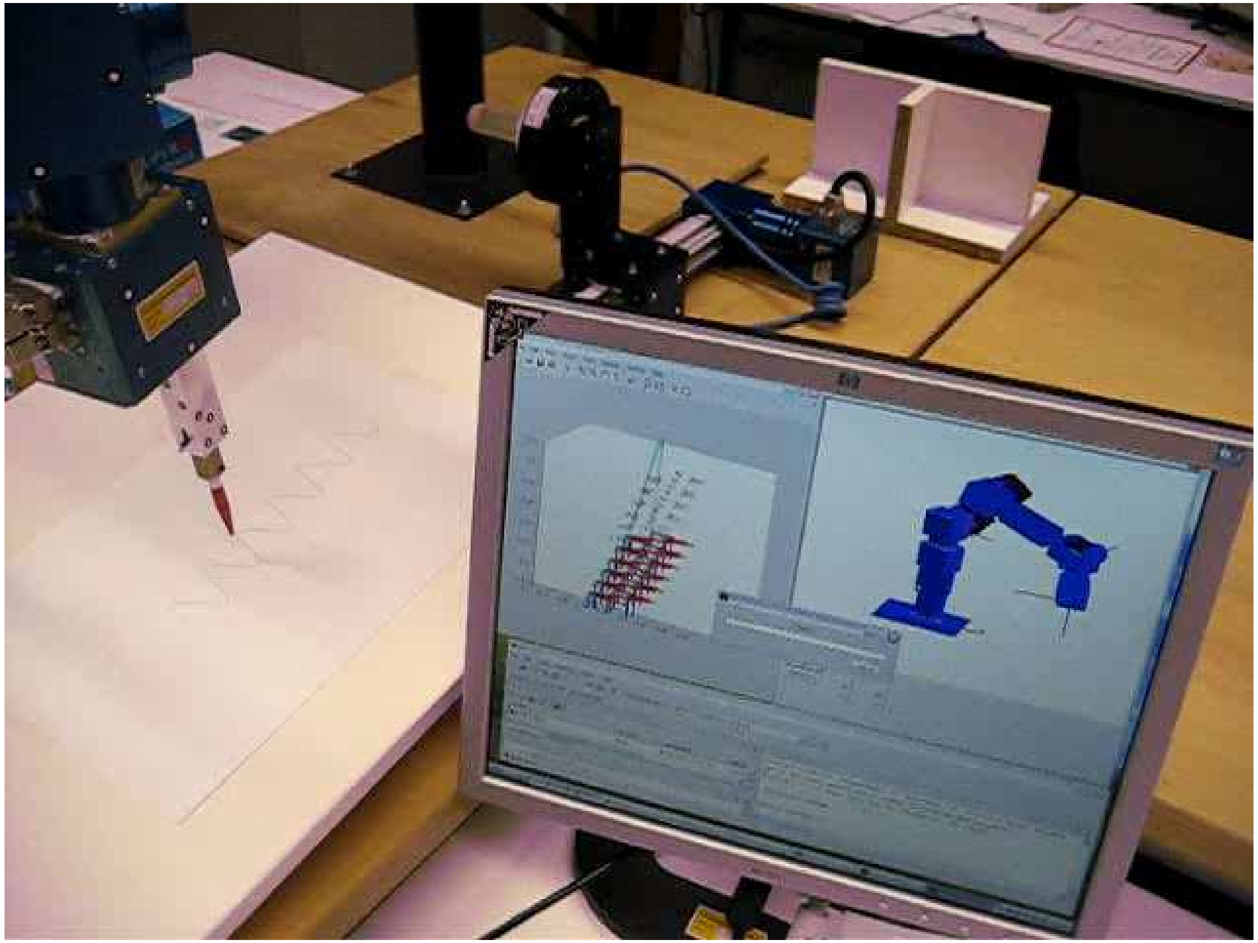
Robot Control *MRobot*

Control Panel:



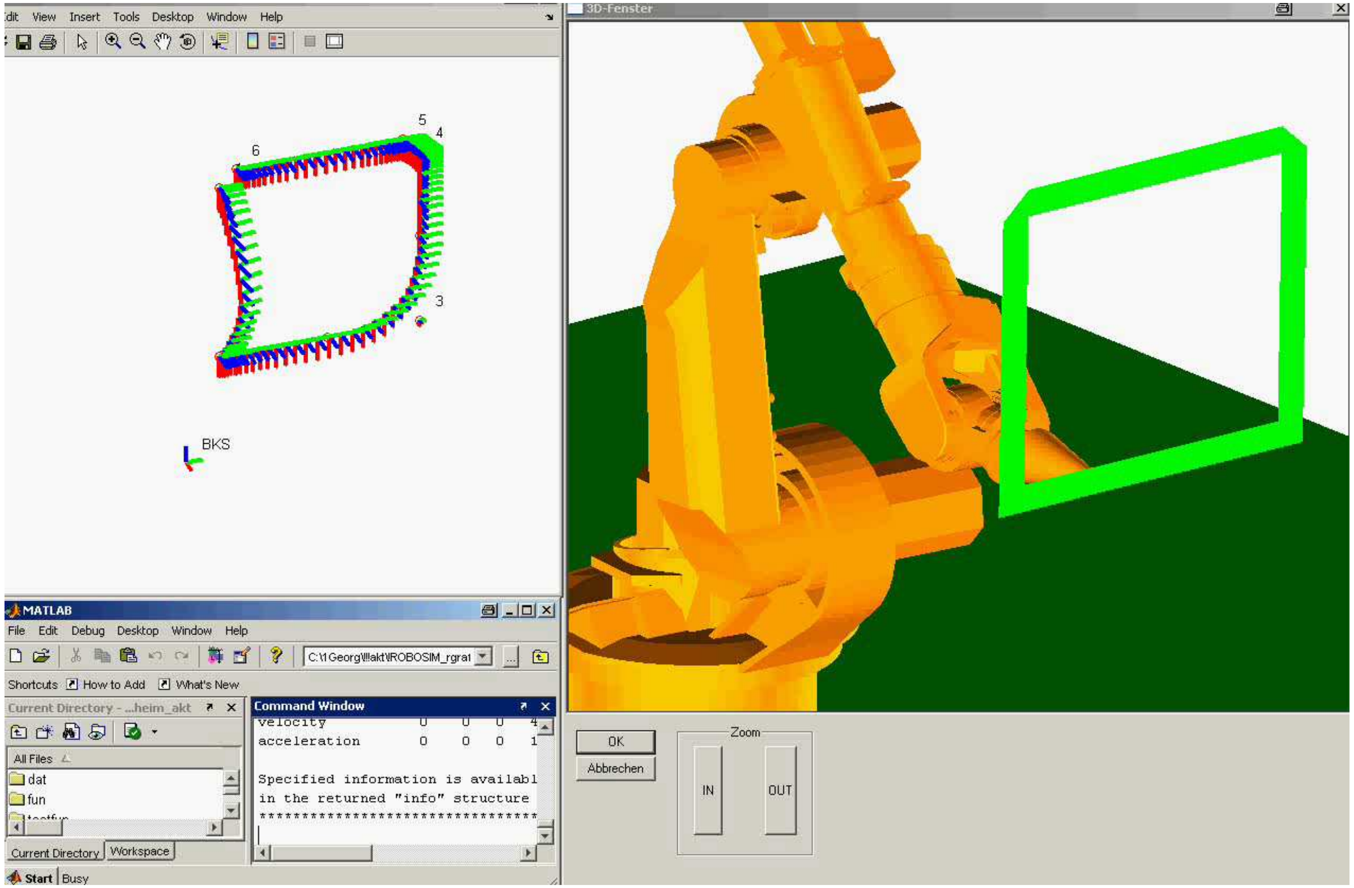
Robot Contol *MRobot*

Synchronous Execution and Simulation



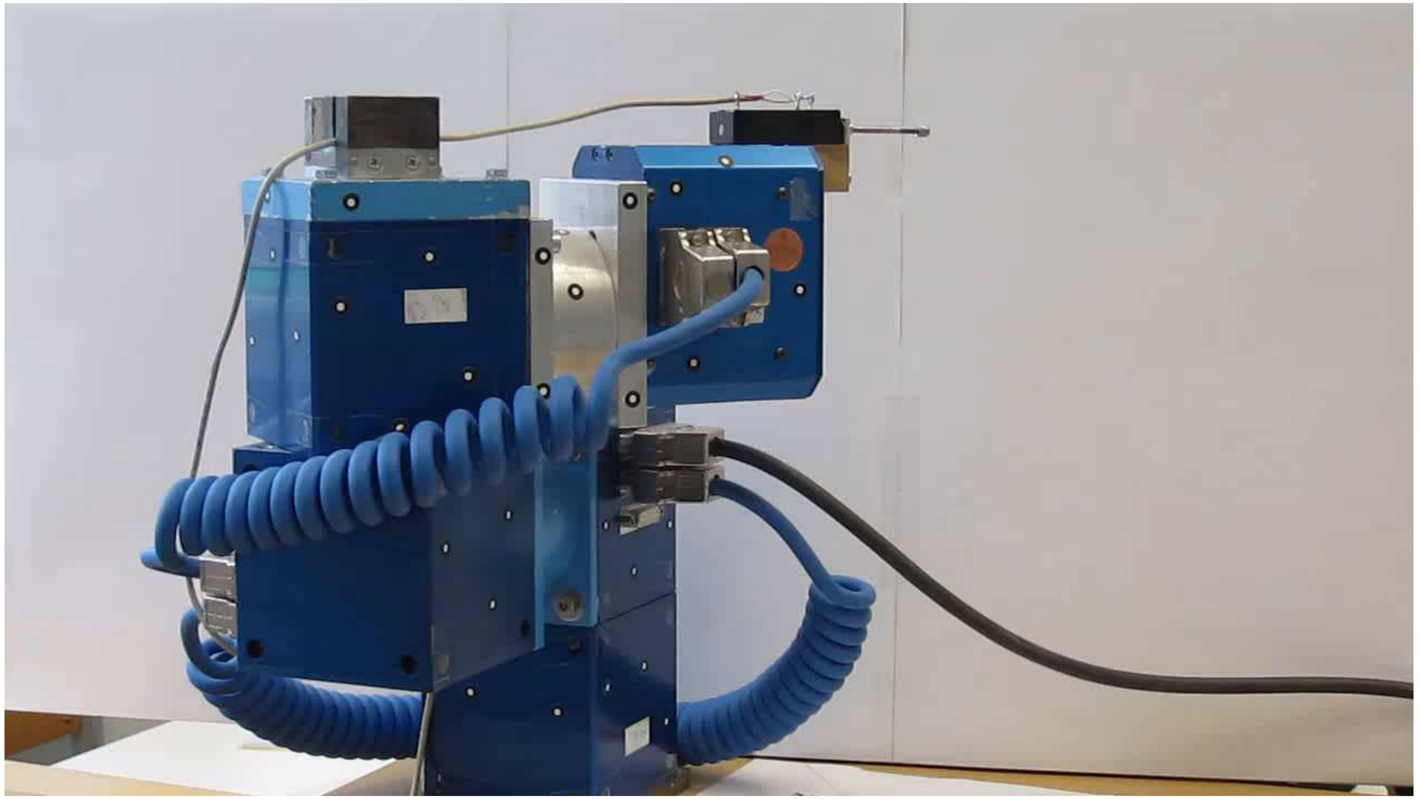
Robot Control *MRobot*

Simulation of KUKA Robot KR15



Robot Control *MRobot*

Motion Control by Distance Sensor



Robot Control *MRobot*

Costeffective Collision Detection by 3D-Webcam

Robot Control *MRobot*

Object Tracing by 3D-Webcam

Robot Contol *MRobot*

Controlling Lightweight Robot of Schunk Company

Laboratory CIM & Robotik

Internet Presentation

- Book: Robotik mit MATLAB:

  http://www.hs-augsburg.de/stark/robotik_mit_matlab/

- MATLAB User Story:

  http://www.mathworks.de/company/user_stories/userstory20581.html

- Laboratory CIM & Robotik:

  http://www.hs-augsburg.de/campus/rotes_tor/j-bau/j3/j307/index.html