



Introduction to Robotics

Lecture 8

Shuang Li, Jianwei Zhang
[sli, zhang]@informatik.uni-hamburg.de



University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

June 12, 2020





Introduction

Spatial Description and Transformations

Forward Kinematics

Robot Description

Inverse Kinematics for Manipulators

Instantaneous Kinematics

Trajectory Generation 1

Trajectory Generation 2

Dynamics

Robot Control

- Introduction

- Internal Sensors of Robots

- PID controller

- Classification of Robot Arm Controllers





Outline (cont.)

- Task-Level planning and Motion planning
- Task-Level planning and Motion planning
- Architectures of Sensor-based Intelligent Systems
- Summary
- Conclusion and Outlook

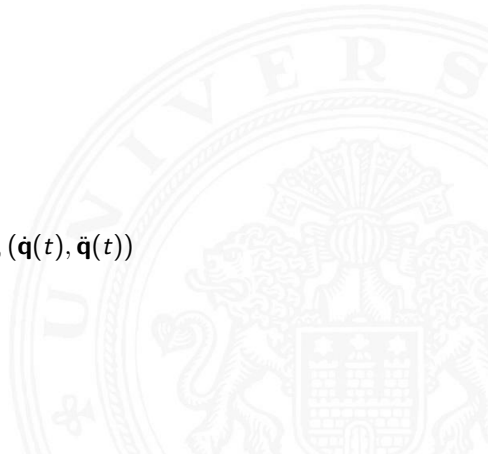




- ▶ **Forward dynamics:**
 - ▶ *Input:* joint forces / torques;
 - ▶ *Output:* kinematics;
 - ▶ *Application:* Simulation of a robot model.
- ▶ **Inverse Dynamics:**
 - ▶ *Input:* desired trajectory of a manipulator;
 - ▶ *Output:* required joint forces / torques;
 - ▶ *Application:* model-based control of a robot.

$\tau(t) \rightarrow$ direct dynamics $\rightarrow \mathbf{q}(t), (\dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))$

$\mathbf{q}(t) \rightarrow$ inverse dynamics $\rightarrow \tau(t)$





General inverse dynamic equations of a manipulator:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$$

Forward dynamic equations of a manipulator:

$$\ddot{\Theta} = M^{-1}(\Theta)(\tau - V(\Theta, \dot{\Theta}) - G(\Theta))$$

$$\dot{\Theta} = \int \ddot{\Theta} dt$$

$$\Theta = \int \dot{\Theta} dt$$

Unlike kinematics, the inverse dynamics is easier to solve than forward dynamics.

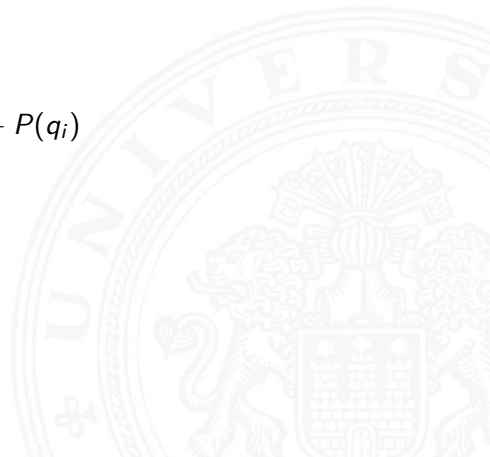


Two methods for calculation:

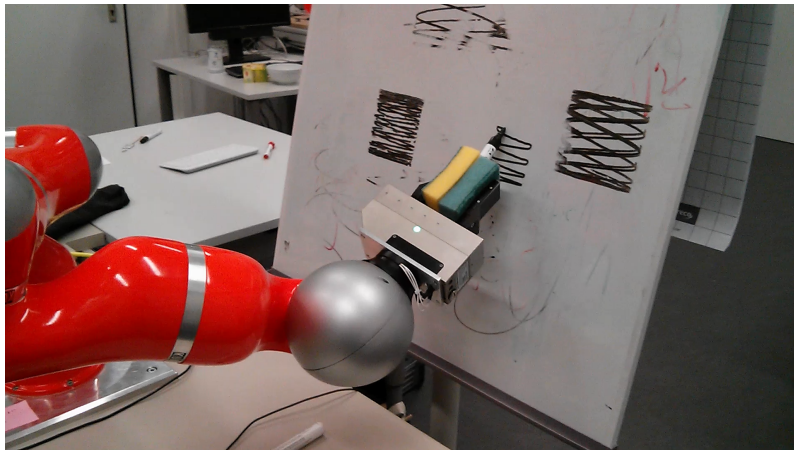
- ▶ Analytical methods
 - ▶ based on Lagrangian equations

$$L(q_i, \dot{q}_i) = K(q_i, \dot{q}_i) - P(q_i)$$

- ▶ Synthetic methods:
 - ▶ based on the Newton-Euler equations

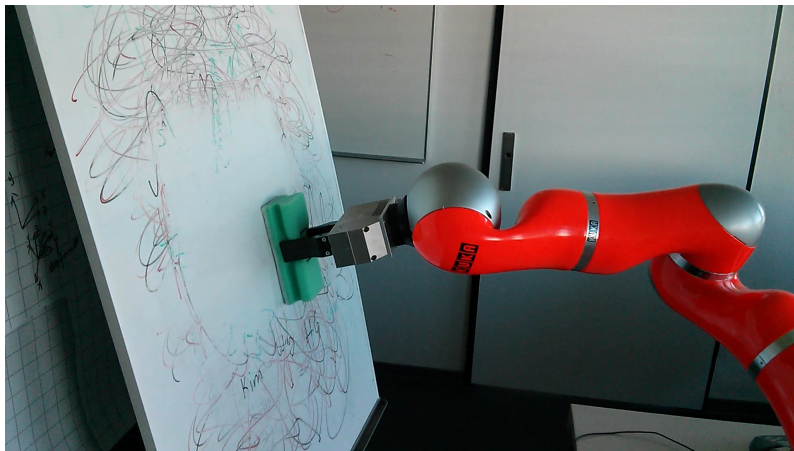


Drawing task





Wiping task



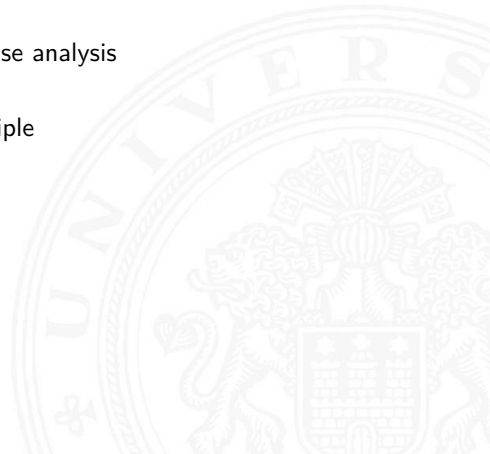


Controller

- ▶ Influences one or more physical variables
 - ▶ meet a control variable
 - ▶ reduce disturbances
- ▶ Compares actual value to reference value
 - ▶ minimize control deviation



- 1788 J. Watt: engine speed governor
- 1877 J. Routh: differential equation for the description of control processes
- 1885 A. Hurwitz: stability studies
- 1932 A. Nyquist: frequency response analysis
- 1940 W. Oppelt: frequency response analysis, Control Engineering becomes an independent discipline
- 1945 H. Bode: discipline new methods for frequency response analysis
- 1950 N. Wiener: statistical methods
- 1956 L. Pontrjagin: optimal control theory, maximum principle
- 1957 R. Bellmann: dynamic programming
- 1960 direct digital control
- 1965 L. Zadeh: Fuzzy-Logic
- 1972 Microcomputer use
- 1975 Control systems for automation
- 1980 Digital device technology
- 1985 Fuzzy-controller for industrial use
- 1995 Artificial neuronal networks for industrial use





Given: dynamic system (to be controlled)

- ▶ Model describing dynamic system (e.g. Jacobian)
- ▶ Input variables – control variables
 - ▶ measured values (sensor data)
- ▶ Output variables – controlled variables
 - ▶ system input (force/torque data)

Problem

- ▶ Keep control variable values constant **and / or**
- ▶ Follow a reference value **and / or**
- ▶ Minimize the influence of disturbances



Sought: controller (for dynamic system)

- ▶ Implement hardware or software controller
- ▶ Alter controlled-variables (output)
- ▶ Based on control variables (input)
- ▶ Solve the problem

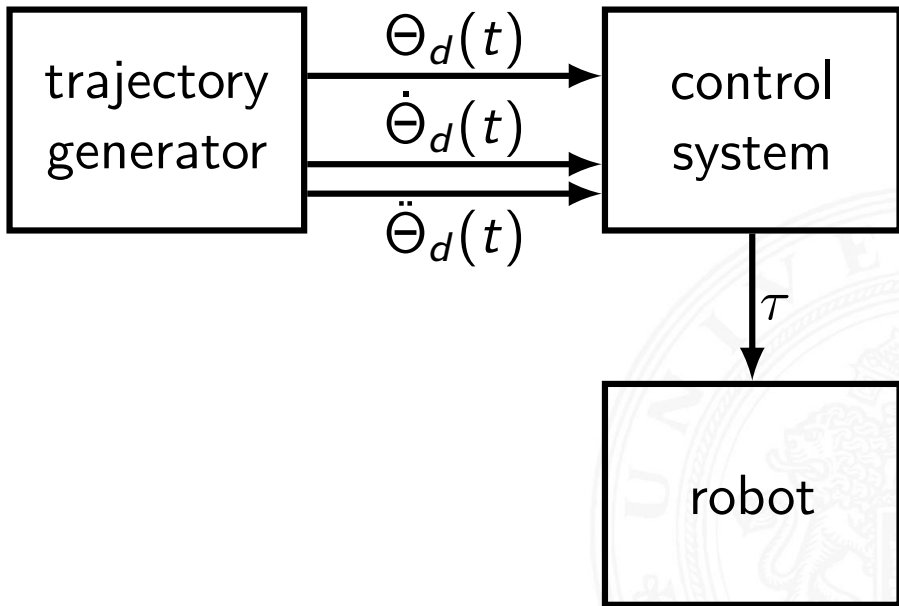


Input

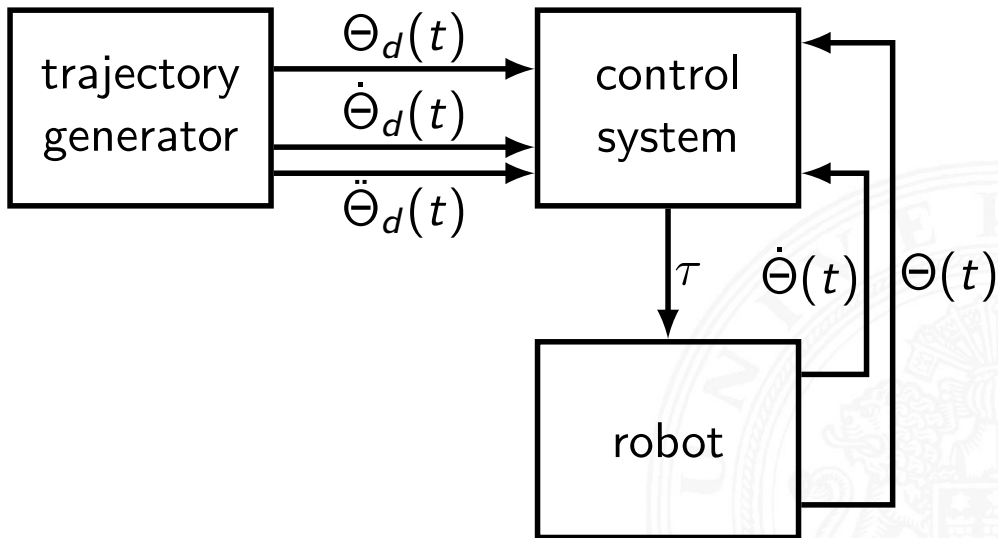
- ▶ Speed over ground
- ▶ Relative speed to traffic
- ▶ Distance to car in front
- ▶ Distance to car behind
- ▶ Weather conditions
- ▶ Relative position in road lane
- ▶ ...

Output

- ▶ Throttle
- ▶ Brakes
- ▶ Steering



Control System of a Robot (cont.)

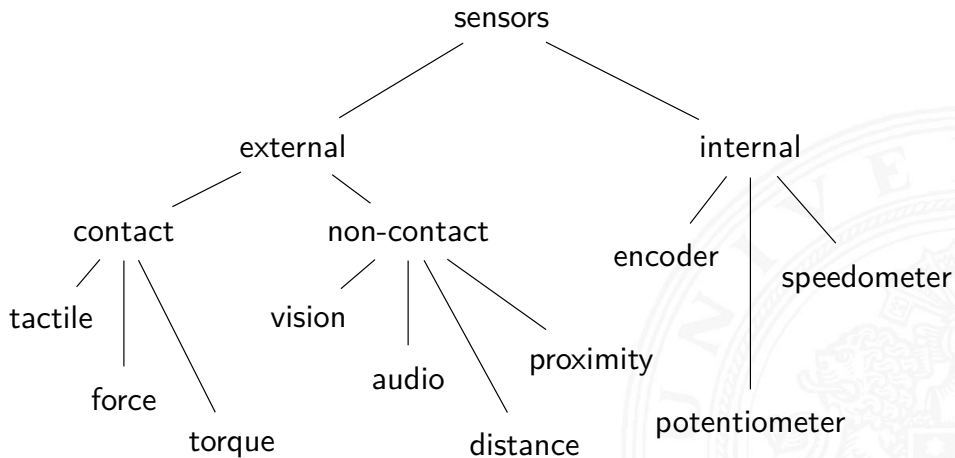




- ▶ Target values
 - ▶ $\Theta_d(t)$
 - ▶ $\dot{\Theta}_d(t)$
 - ▶ $\ddot{\Theta}_d(t)$
- ▶ Magnitude of error
 - ▶ $E = \Theta_d - \Theta, \dot{E} = \dot{\Theta}_d - \dot{\Theta}$
- ▶ Output (Control) value
 - ▶ $\Theta(t)$
 - ▶ $\dot{\Theta}(t)$
- ▶ Controlled value
 - ▶ τ



Sensor Classification Hierarchy





- ▶ Placed inside the robot
- ▶ Monitor the internal state of the robot
 - ▶ e.g. position and velocity of a joint

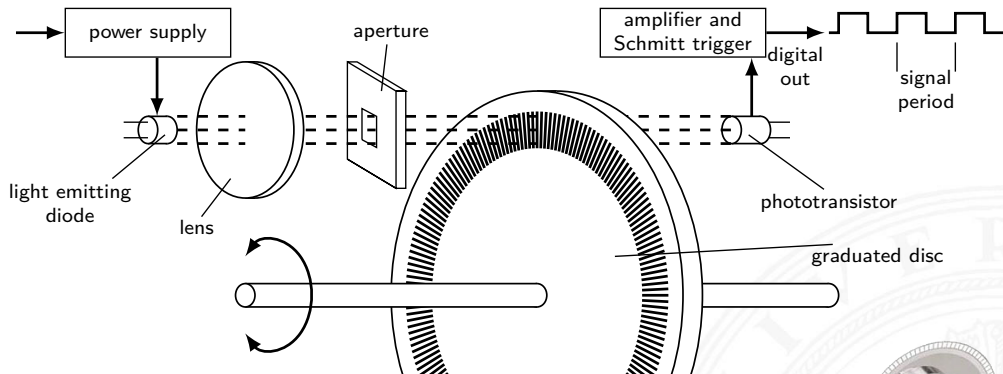
Position measurement systems

- ▶ Potentiometer
- ▶ Incremental/absolute encoder
- ▶ Resolver

Velocity measurement systems

- ▶ Speedometers
- ▶ Calculate from position change over time

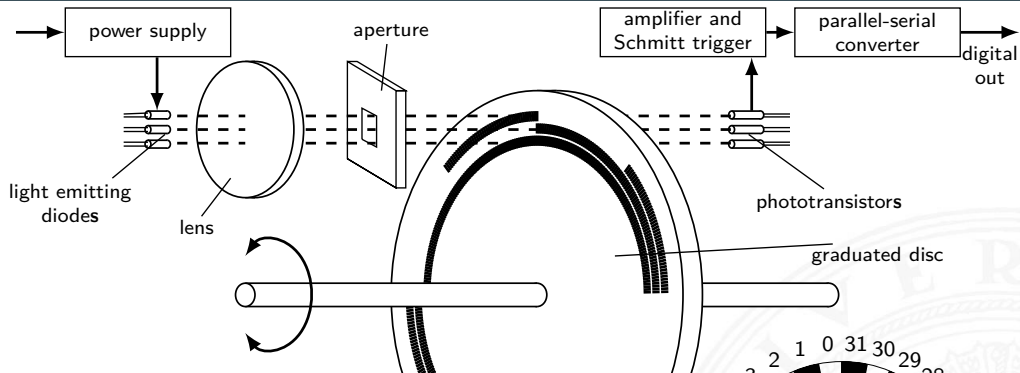
Optical Incremental Encoders



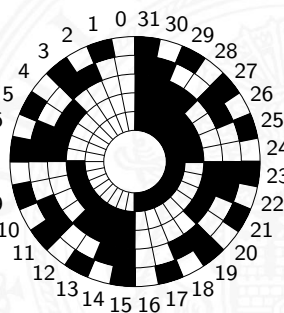
- ▶ An optical encoder reads the lines
- ▶ The disc is mounted to the shaft of the joint motor
 - ▶ PUMA-560: 1:1 ratio; .0001 rad/bit accuracy
- ▶ one special line is marked as the “zero-position”

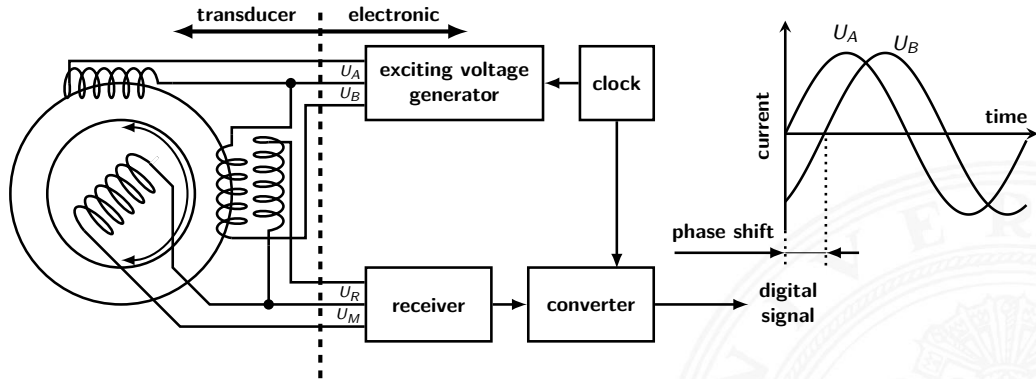


Optical Absolute Encoder



- ▶ multiple LEDs and phototransistors
- ▶ e.g. 5 bit dual code gives 32 angular positions and 11.25° resolution
- ▶ parallel-serial converter required
- ▶ absolute positioning and direction encoding

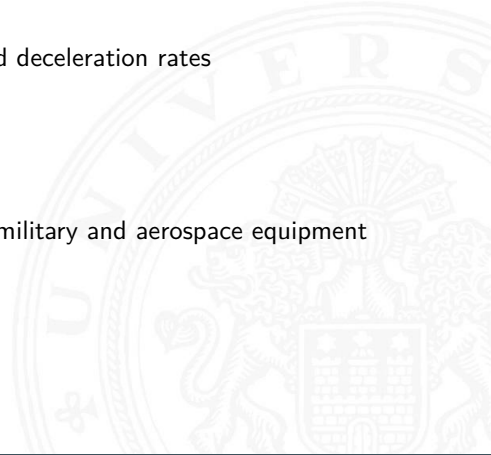




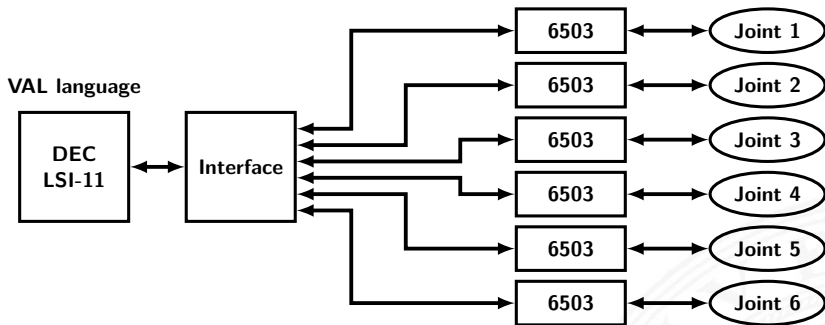
- ▶ analog rotation encoding
- ▶ phase shift between U_A and U_B determines rotation
- ▶ precision depending on digital converter



- ▶ Encoder:
 - ▶ higher accuracy
 - ▶ simplicity of integration, and update
 - ▶ suitable for applications with high acceleration and deceleration rates
- ▶ Resolver:
 - ▶ lack of sensitive optics
 - ▶ resistant to electrical disturbances
 - ▶ complexity of integrating a resolver into a system
 - ▶ suitable for extremely harsh applications, such as military and aerospace equipment

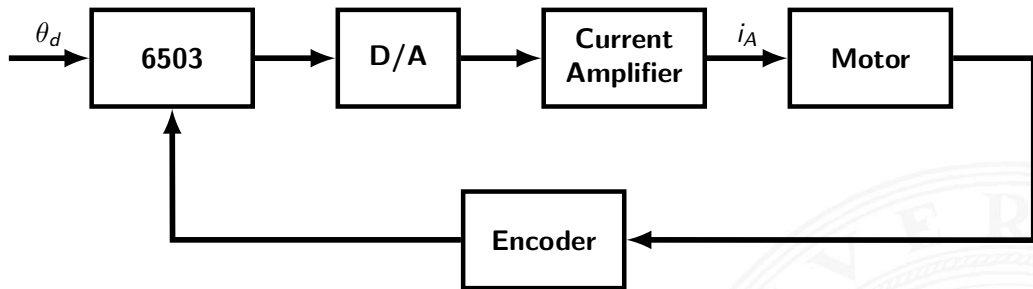


Control System Architecture of PUMA-Robot



- ▶ two-level hierachical structure of control system
- ▶ *DEC LSI-11* sends joint values at 35.7 Hz (28 ms)
 - ▶ trajectory
- ▶ Distance of actual value to goal value is interpolated
 - ▶ using 8,16,**32** or 64 increments

Control System Architecture of PUMA-Robot (cont.)



- ▶ The joint control loop operates at 1143 Hz (0.875 ms)
- ▶ Encoders are used as position sensors
- ▶ No dedicated speedometer
 - ▶ velocity is calculated as the difference of joint positions over time

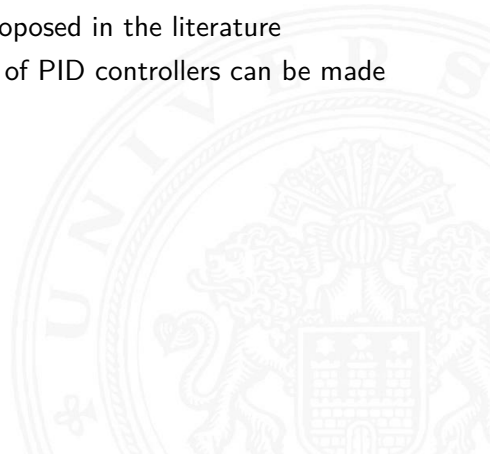


- ▶ more than half of the industrial controllers in use today are PID controllers or modified PID controllers
- ▶ many different types of tuning rules have been proposed in the literature
- ▶ Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site

P Proportional controller

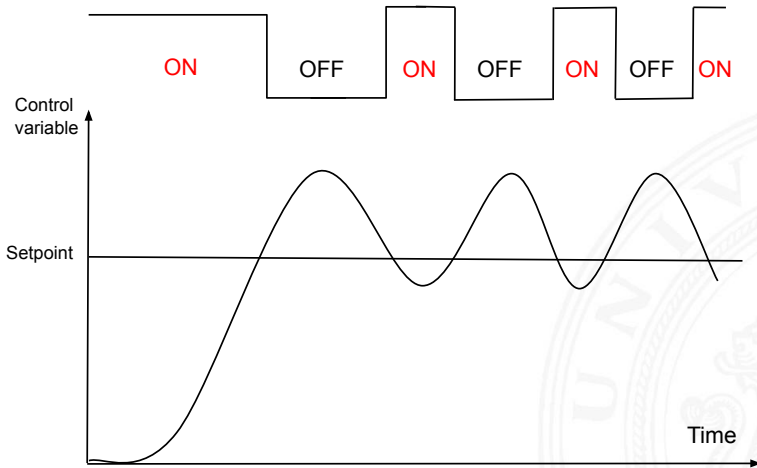
I Integral controller

D Derivative controller



Bang Bang (On-off) controller

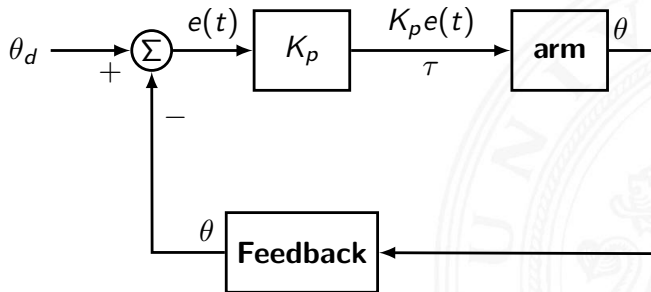
This is the simplest form of control.





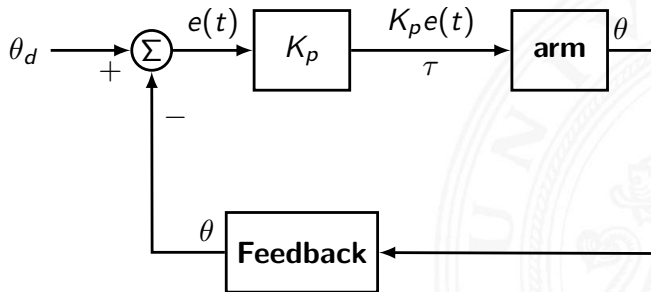
In proportional mode, there is a continuous linear relation between value of the controlled variable and position of the final control element.

- ▶ $e(t) = \theta_d - \theta$
- ▶ output of proportional controller is $\tau(t) = K_p e(t)$, K_p is proportional gain.



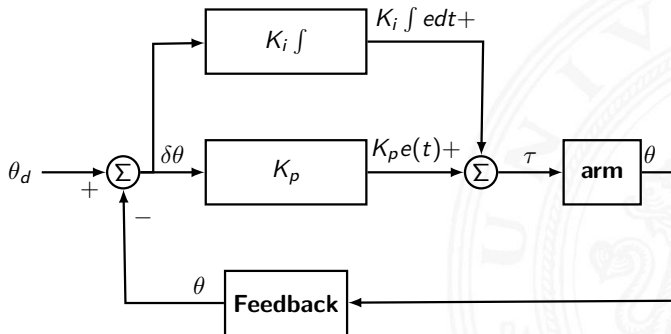


- ▶ Using P control is simple, but often insufficient:
 - ▶ If K_p is small, the sensor reading will approach the setpoint slowly and never reach it
 - ▶ As the gain is increased the system responds faster to changes in set-point but becomes progressively underdamped and eventually unstable.
 - ▶ If K_p is large, the system may overshoot, oscillate (i.e. become unstable)



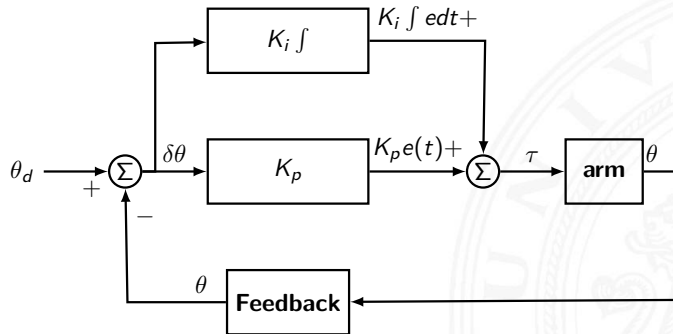
Proportional-Integral (PI) control

- ▶ $\tau(t) = K_p e(t) + K_i \int e dt$
- ▶ The P term will take care of the large movement
- ▶ Integral signal is sum of all instantaneous errors
- ▶ The I term will take care of any steady-state error





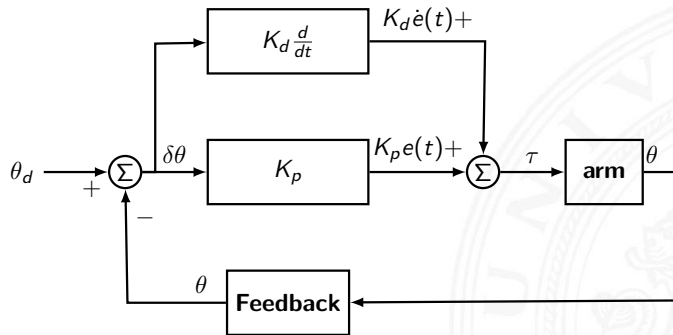
- ▶ It eliminates steady-state error
- ▶ It can help with stability of the system, especially if K_p is large
- ▶ But, it responds relatively slowly to an error signal





Proportional-Derivative (PD) control

- ▶ $\tau(t) = K_p e(t) + K_d \dot{e}(t)$
- ▶ Differential term at time $n = K_d(e(n) - e(n - 1))/\Delta t$

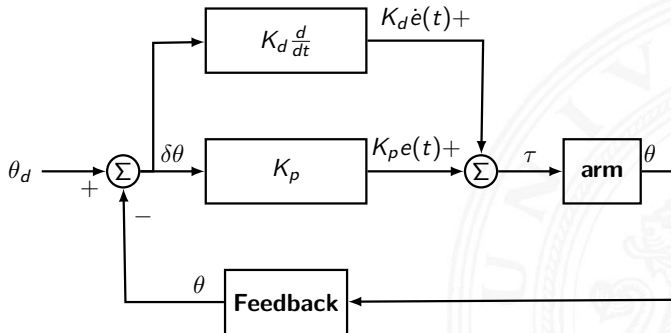




Proportional-Derivative (PD) control (cont.)

The main advantages of the PD controllers are:

- ▶ The derivative term acts as "brake" to the system
- ▶ It can improve the system's tolerance to external disturbances



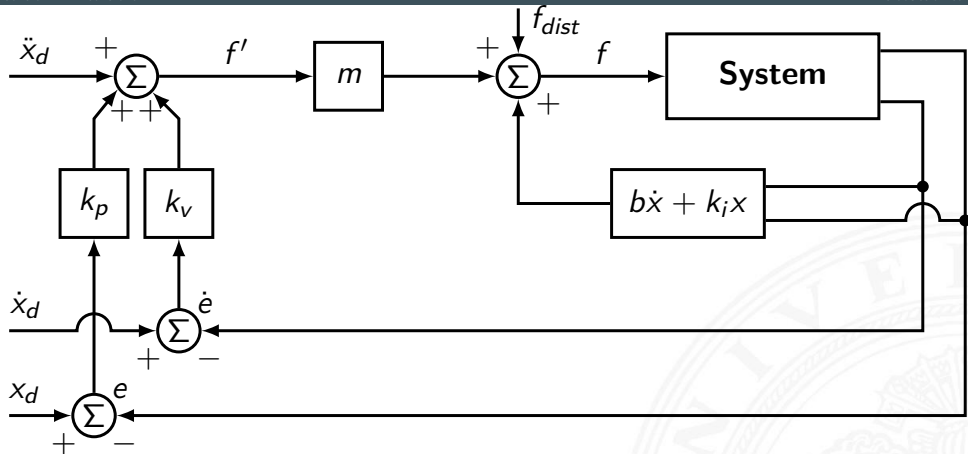


- P** Proportional controller: $\tau(t) = k_p \cdot e(t)$
The amplification factor k_p defines the sensitivity.
- I** Integral controller: $\tau(t) = k_i \cdot \int_{t_0}^t e(t') dt'$
Long term errors will sum up.
- D** Derivative controller: $\tau(t) = k_v \cdot \dot{e}(t)$
This controller is sensitive to changes in the deviation.

Combined \Rightarrow PID-controller:

$$\tau(t) = k_p \cdot e(t) + k_v \cdot \dot{e}(t) + k_i \int_{t_0}^t e(t') dt'$$

Linear Control for Trajectory Tracking



$$f' = \ddot{x}_d + k_v \dot{e} + k_p e + k_i \int e dt \quad (47)$$

is called the principle of PID-control.

Summary: the characteristics of P, I, D controllers

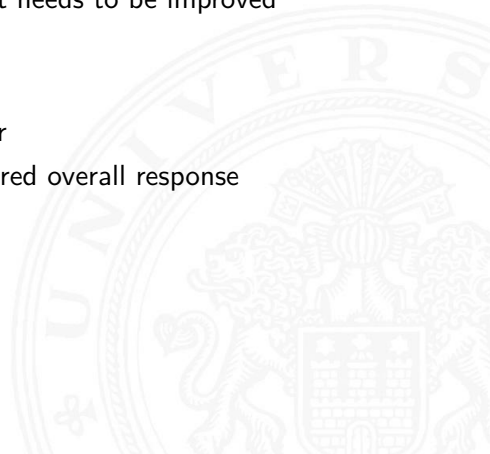
	Rise time	Overshoot	Settling time	S-S error
K_p	decrease	increase	small change	decrease
K_i	decrease	increase	increase	eliminate
K_d	small change	decrease	decrease	small change

Further Resources

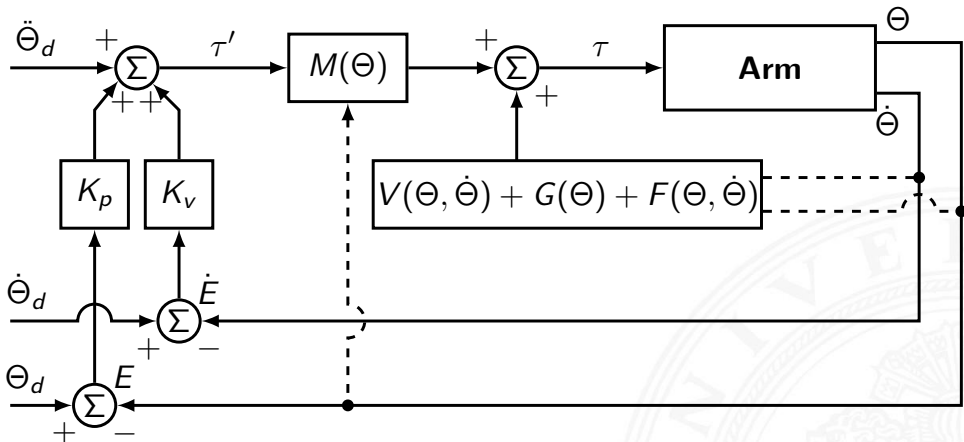
- ▶ PID Control with Python (simple-pid)
- ▶ PID Control with MATLAB and Simulink



1. Obtain an open-loop response and determine what needs to be improved
2. Add a P control to improve the rise time
3. Add a D control to improve the overshoot
4. Add a I control to eliminate the steady-state error
5. Adjust each of K_p , K_d , K_i until you obtain a desired overall response



Model-Based Control for Trajectory Tracking



The dynamic equation:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)$$

where $M(\Theta)$ is the position-dependent $n \times n$ -mass matrix of the manipulator, $V(\Theta, \dot{\Theta})$ is a $n \times 1$ -vector of centripetal and Coriolis factors, and $G(\Theta)$ is a complex function of Θ , the position of all joints of the manipulator.



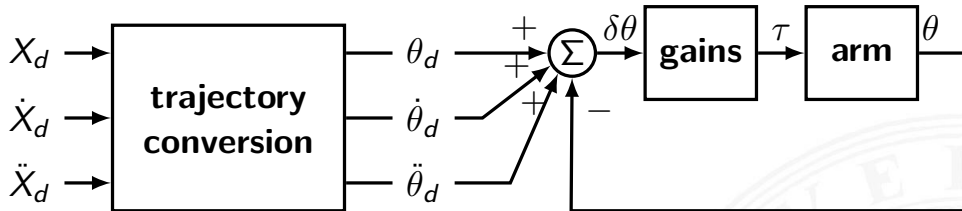
As the problem of trajectory-tracking:

- ▶ Joint space: PID, plus model-based
- ▶ Cartesian space: joint-based
 - ▶ using kinematics or using inverse Jacobian calculation
- ▶ Adaptive: model-based adaptive control, self-tuning
 - ▶ controller (structure and parameter) adapts to the time-invariant or unknown system-behavior
 - ▶ basic control circle is superimposed by an adaptive system
 - ▶ process of adaption consists of three phases
 - ▶ identification
 - ▶ decision-process
 - ▶ modification
- ▶ Hybrid force and position control is also a popular research topic



Control in Cartesian Space – Method I

Joint-based control with Cartesian trajectory input

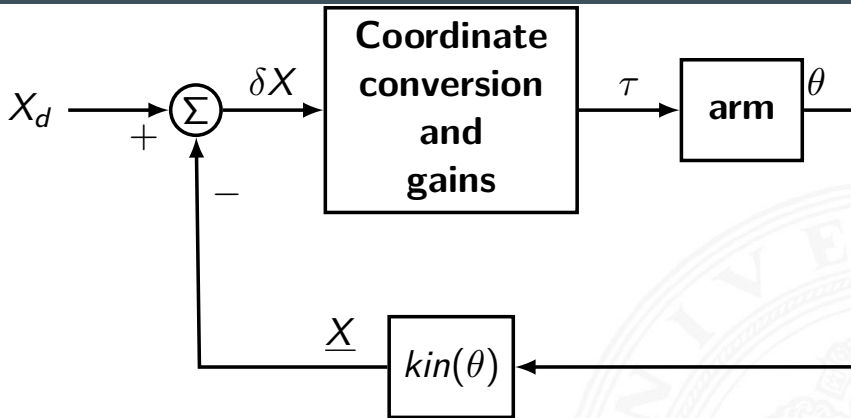


- ▶ Cartesian trajectory is converted into joint space first
- ▶ joint space trajectory is sent to the controller
- ▶ trajectory controller sends joint targets to motor controllers
- ▶ motor controller sends torque data to motor
- ▶ sensors output joint state



Control in Cartesian Space – Method II

Cartesian control via calculation of kinematics

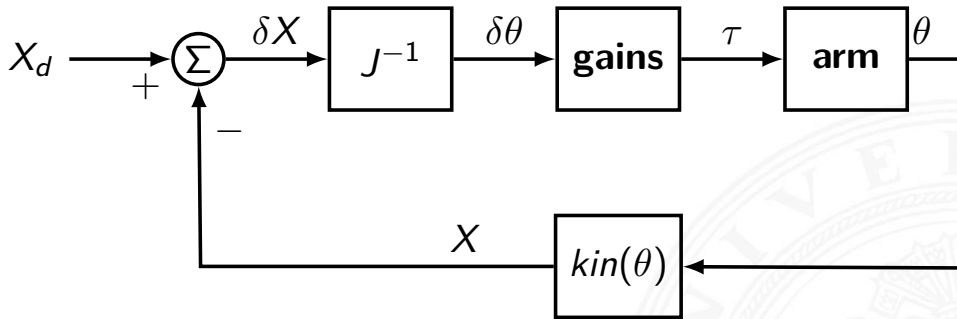


- ▶ controller operates in cartesian space
- ▶ joint space conversion within control cycle
- ▶ error values in cartesian space using FK



Control in Cartesian Space – Method III

Cartesian control via calculation of inverse Jacobian



- ▶ no explicit joint space conversion
- ▶ dynamic conversion using inverse Jacobian

Scientific Research

- ▶ model-based control
- ▶ adaptive control
- ▶ hybrid control

Industrial robots

- ▶ PID-control system with gravity compensation

$$\tau = \dot{\Theta}_d + K_v \dot{E} + K_p E + K_i \int E dt + \hat{G}(\Theta)$$



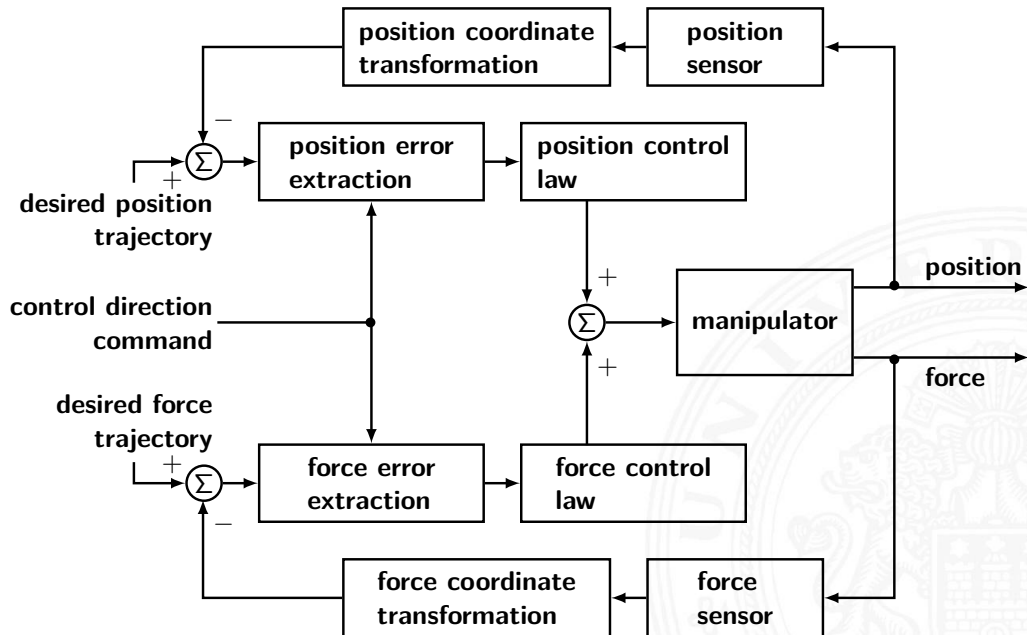
Motivation

Certain tasks require control of both: position and force of the end effector:

- ▶ assembly
- ▶ grinding
- ▶ opening/closing doors
- ▶ crank winding
- ▶ ...

An example shows two feedback loops for separate control of position and force

Hybrid Control of Force and Position (cont.)





Franke Emika Panda





- [1] G.-Z. Yang, R. J. Full, N. Jacobstein, P. Fischer, J. Bellingham, H. Choset, H. Christensen, P. Dario, B. J. Nelson, and R. Taylor, “Ten robotics technologies of the year,” 2019.
- [2] J. K. Yim, E. K. Wang, and R. S. Fearing, “Drift-free roll and pitch estimation for high-acceleration hopping,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8986–8992, IEEE, 2019.
- [3] J. F. Engelberger, *Robotics in service*. MIT Press, 1989.
- [4] K. Fu, R. González, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill series in CAD/CAM robotics and computer vision, McGraw-Hill, 1987.
- [5] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*. Artificial Intelligence Series, MIT Press, 1981.
- [6] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*. Always learning, Pearson Education, Limited, 2013.



- [7] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.
- [8] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2009.
- [9] W. Böhm, G. Farin, and J. Kahmann, "A Survey of Curve and Surface Methods in CAGD," *Comput. Aided Geom. Des.*, vol. 1, pp. 1–60, July 1984.
- [10] J. Zhang and A. Knoll, "Constructing Fuzzy Controllers with B-spline Models - Principles and Applications," *International Journal of Intelligent Systems*, vol. 13, no. 2-3, pp. 257–285, 1998.
- [11] M. Eck and H. Hoppe, "Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*, (New York, NY, USA), pp. 325–334, ACM, 1996.



- [12] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*. PhD thesis, Bielefeld University, 2001.
- [13] J. H. Reif, "Complexity of the Mover's Problem and Generalizations - Extended Abstract," *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pp. 421–427, 1979.
- [14] J. T. Schwartz and M. Sharir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artificial Intelligence*, vol. 37, no. 1, pp. 157–169, 1988.
- [15] J. Canny, *The Complexity of Robot Motion Planning*. MIT press, 1988.
- [16] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*. Cambridge, MA, USA: MIT Press, 1992.
- [17] O. Khatib, "The Potential Field Approach and Operational Space Formulation in Robot Control," in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.



- [18] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, T.-Y. Li, and P. Raghavan, “A Random Sampling Scheme for Path Planning,” in *Robotics Research* (G. Giralt and G. Hirzinger, eds.), pp. 249–264, Springer London, 1996.
- [19] R. Geraerts and M. H. Overmars, “A Comparative Study of Probabilistic Roadmap Planners,” in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.
- [20] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue, “The Experimental Humanoid Robot H7: A Research Platform for Autonomous Behaviour,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 79–107, 2007.
- [21] R. Brooks, “A robust layered control system for a mobile robot,” *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.
- [22] M. J. Mataric, “Interaction and intelligent behavior.,” tech. rep., DTIC Document, 1994.
- [23] M. P. Georgeff and A. L. Lansky, “Reactive reasoning and planning.,” in *AAAI*, vol. 87, pp. 677–682, 1987.



- [24] J. Zhang and A. Knoll, *Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control*, pp. 367–385.
Heidelberg: Physica-Verlag HD, 2001.
- [25] J. S. Albus, “The nist real-time control system (rcs): an approach to intelligent systems research,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 157–174, 1997.
- [26] A. Meystel, “Nested hierarchical control,” 1993.
- [27] G. Saridis, “Machine-intelligent robots: A hierarchical control approach,” in *Machine Intelligence and Knowledge Engineering for Robotic Applications* (A. Wong and A. Pugh, eds.), vol. 33 of *NATO ASI Series*, pp. 221–234, Springer Berlin Heidelberg, 1987.
- [28] T. Fukuda and T. Shibata, “Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network,” in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1, pp. 269–274 vol.1, Jun 1992.
- [29] R. C. Arkin and T. Balch, “Aura: principles and practice in review,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.



- [30] E. Gat, "Integrating reaction and planning in a heterogeneous asynchronous architecture for mobile robot navigation," *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 70–74, 1991.
- [31] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration*.
Master thesis, Universität Hamburg, 2015.
- [32] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual*.
Addison-Wesley Pub. Co., 1986.
- [33] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter*.
Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.
- [34] R. Schilling, *Fundamentals of robotics: analysis and control*.
Prentice Hall, 1990.
- [35] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*.
Cambridge, MA, USA: MIT Press, 1990.



- [36] M. Spong, *Robot Dynamics And Control*.
Wiley India Pvt. Limited, 2008.

