Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN Faculty
Department of Informatics

Introduction to Robotics

Lecture 7

Zhaopeng Chen, Shuang Li, Jianwei Zhang
zhaopeng.chen@uni-hamburg.de, [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 05, 2020

Outline

Outline (cont.)

Dynamics of multibody systems

- A multibody system is a mechanical system of single bodies
  - connected by joints,
  - influenced by forces
- The term `dynamics` describes the behavior of bodies influenced by forces
  - Typical forces: weight, friction, centrifugal, magnetic, spring, . . .
- `kinematics` just models the motion of bodies (without considering forces), therefore it can be seen as a subset of dynamics

Forward and inverse Dynamics

We consider a force F and its effect on a body:

$$F = m \cdot a = m \cdot \dot{v}$$

In order to solve this equation, two of the variables need to be known.

Forward Dynamics

If the force F and the mass of the body m is known:

$$a = \dot{v} = \frac{F}{m}$$

Hence the following can be determined:

- velocity (by integration)
- coordinates of single bodies
- forward dynamics
- mechanical stress of bodies

Forward Dynamics (cont.)

Input

τ_i = torque at joint i that effects a trajectory Θ.
$i = 1, \ldots, n$, where n is the number of joints.

Output

Θ_i = joint angle of i
$\dot{\Theta}_i$ = angular velocity of joint i
$\ddot{\Theta}_i$ = angular acceleration of joint i

Inverse Dynamics

If the time curves of the joint angles are known, it can be differentiated twice.

This way,

- internal forces
- and torques

can be obtained for each body and joint.

Problems of highly dynamic motions:

- models are not as complex as the real bodies
- differentiating twice (on sensor data) leads to high inaccuracy

Inverse Dynamics (cont.)

Input

Θ_i = joint angle i
$\dot{\Theta}_i$ = angular velocity of joint i
$\ddot{\Theta}_i$ = angular acceleration of joint i
$i = 1, \ldots, n$, where n is the number of joints.

Output

τ_i = required torque at joint i to produce trajectory Θ.

Dynamics of Manipulators

- **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.

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

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

Dynamics of Manipulators (cont.)

Two methods for calculation:

- Analytical methods
  - based on Lagrangian equations
- Synthetic methods:
  - based on the Newton-Euler equations

Computation time

Complexity of solving the Lagrange-Euler-model is $O(n^4)$ where n is the number of joints.

$n = 6$: 66,271 multiplications and 51,548 additions.

Lagrangian equations

The description of manipulator dynamics is directly based on the relations between the kinetic K and potential energy P of the manipulator joints.

Here:

- constraining forces are not considered
- deep knowledge of mechanics is necessary
- high effort of defining equations
- can be solved by software

Lagrangian Equations

The Lagrangian function L is defined as the difference between kinetic energy K and potential energy P of the system.

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

- K: kinetic energy due to linear velocity of the link's center of mass and angular velocity of the link
- P: potential energy stored in the manipulator that is the sum of the potential energy in the individual links

Lagrangian Equations (cont.)

The Lagrangian function L is defined as:

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

Theorem

The motion equations of a mechanical system with coordinates $\mathbf{q} \in \Theta^n$ and the Lagrangian function L is defined by:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = F_i, \quad i = 1, \dots, n$$

where
q_i: the coordinates, where the kinetic and potential energy is defined;
$\dot{q}_i$: the velocity;
F_i: the force or torque, depending on the type of joint (rotational or linear)

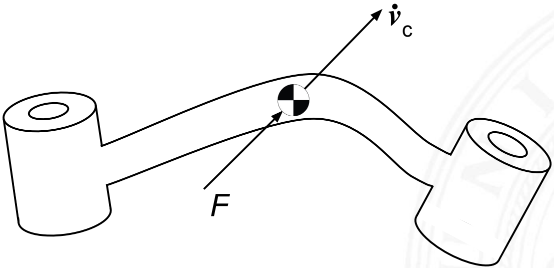
Recursive Newton-Euler Method

- Determine the kinematics from the fixed base to the TCP (relative kinematics)
- The resulting acceleration leads to forces towards rigid bodies
- The combination of constraining forces, payload forces, weight forces and working forces can be defined for every rigid body. All torques and momentums need to be in balance
- Solving this formula leads to the joint forces
- Especially suitable for serial kinematics of manipulator

1. Newton's equation

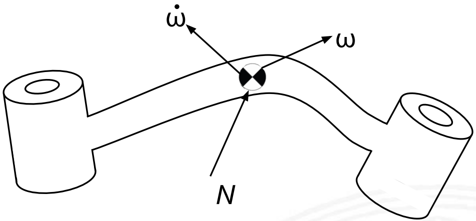
$$F = m\dot{v}_c$$

where F is the force acting at the center of mass of a body, m is the total mass of the body, v_c is the acceleration.



2. Euler's equation

$$\tau = {}^{C}\mathbf{I}\dot{\omega} + \omega \times {}^{C}\mathbf{I}\omega$$



- where ${}^{C}\mathbf{I}$ is the inertia tensor of the body written in a frame C, whose origin is located at the center of the mass.

$$^{C}I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & Izz \end{bmatrix}$$

- τ is the torque
- $\omega, \dot{\omega}$ are the angular velocity and angular acceleration respectively

Influencing factors to robot dynamics

- Functional affordance
  - trajectory and velocity of links
  - load on a link
- Control quantity
  - velocity and acceleration of joints
  - forces and torques
- Robot-specific elements
  - geometry
  - mass distribution

Aim of determining robot dynamics

- Determining joint forces and torques for one point of a trajectory $(\Theta, \dot{\Theta}, \ddot{\Theta})$
- Determining the motion of a link or the complete manipulator for given joint-forces and -torques (τ)

To achieve this the mathematical model is applied.

Formulation of robot dynamics

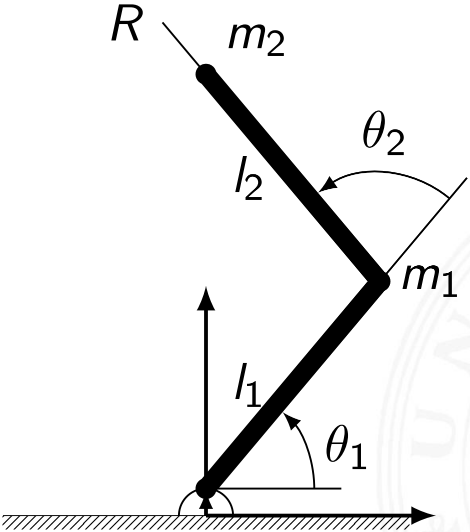
- Combining the different influence factors in the robot specific motion equation from kinematics $(\Theta, \dot{\Theta}, \ddot{\Theta})$
- Practically the Newton-, Euler- and motion-equation for each joint are combined
- Advantages: numerically efficient, applicable for complex geometry, can be modularized

Interim Conclusions

- We can determine the forces with the Newton-equation
- The Euler-equation provides the torque
- **The combination provides force and torque for each joint.**

Example: A 2 DOF manipulator

Dynamics of a multibody system, example: a two joint manipulator.



Newton-Euler-Equations for 2 DOF manipulator

Using Newton's second law, the forces at the center of mass at link 1 and 2 are:

$$\mathbf{F}_1 = m_1 \ddot{\mathbf{r}}_1$$

$$\mathbf{F}_2 = m_2 \ddot{\mathbf{r}}_2$$

where

$$\mathbf{r}_1 = \frac{1}{2} l_1 (\cos\theta_1 \vec{i} + \sin\theta_1 \vec{j})$$

$$\mathbf{r}_2 = 2\mathbf{r}_1 + \frac{1}{2} l_2 [\cos(\theta_1 + \theta_2)\vec{i} + \sin(\theta_1 + \theta_2)\vec{j}]$$

Euler equations:

$$\tau_1 = \mathbf{I}_1 \dot{\omega}_1 + \omega_1 \times \mathbf{I}_1 \omega_1$$

$$\tau_2 = \mathbf{I}_2 \dot{\omega}_2 + \omega_2 \times \mathbf{I}_2 \omega_2$$

where

$$\mathbf{I}_1 = \frac{m_1 {l_1}^2}{12} + \frac{m_1 R^2}{4}$$

$$\mathbf{I}_2 = \frac{m_2 {l_2}^2}{12} + \frac{m_2 R^2}{4}$$

Newton-Euler-Equations for 2 DOF manipulator (cont.)

The angular velocities and angular accelerations are:

$$\omega_1 = \dot{\theta}_1$$

$$\omega_2 = \dot{\theta}_1 + \dot{\theta}_2$$

$$\dot{\omega}_1 = \ddot{\theta}_1$$

$$\dot{\omega}_2 = \ddot{\theta}_1 + \ddot{\theta}_2$$

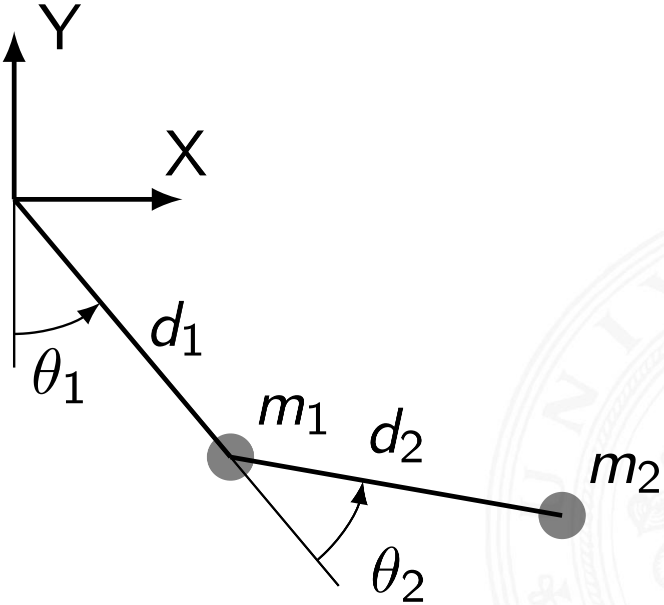
As $\omega_i \times \mathbf{I}_i \omega_i = 0$, the torques at the center of mass of links 1 and 2 are:

$$\tau_1 = \mathbf{I}_1 \ddot{\theta}_1$$

$$\tau_2 = \mathbf{I}_2 (\ddot{\theta}_1 + \ddot{\theta}_2)$$

$\mathbf{F}_1, \mathbf{F}_2, \tau_1, \tau_2$ are used for force and torque balance and are solved for joint 1 and 2.

Example: A two joint manipulator



Langragian Method for two joint manipulator

The kinetic energy of mass m_1 is:

$$K_1 = \frac{1}{2} m_1 \ d_1{}^2 \ \dot{\theta_1}^2$$

The potential energy is:

$$P_1 = -m_1 \ g \ d_1 \ cos(\theta_1)$$

The cartesian positions are:

$$x_2 = d_1 sin(\theta_1) + d_2 sin(\theta_1 + \theta_2)$$
$$y_2 = -d_1 cos(\theta_1) - d_2 cos(\theta_1 + \theta_2)$$

Langragian Method for two joint manipulator (cont.)

The cartesian components of velocity are:

$$\dot{x}_2 = d_1 cos(\theta_1)\dot{\theta}_1 + d_2 cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{y}_2 = d_1 sin(\theta_1)\dot{\theta}_1 + d_2 sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

The square of velocity is:

$${v_2}^2 = {\dot{x}_2}^2 + {\dot{y}_2}^2$$

The kinetic energy of link 2 is:

$$K_2 = \frac{1}{2} m_2 {v_2}^2$$

The potential energy of link 2 is:

$$P_2 = -m_2 g d_1 cos(\theta_1) - m_2 g d_2 cos(\theta_1 + \theta_2)$$

Langragian Method for two joint manipulator (cont.)

The Lagrangian function is:

$$L = (K_1 + K_2) - (P_1 + P_2)$$

The force/torque to joint 1 and 2 are:

$$\tau_1 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1}$$

$$\tau_2 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2}$$

Langragian Method for two joint manipulator (cont.)

τ_1 and τ_2 are expressed as follows:

$$\begin{aligned}\tau_1 =& D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 \\ &+ D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_2\dot{\theta}_1 + D_1\end{aligned}$$

$$\begin{aligned}\tau_2 =& D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 \\ &+ D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_2\dot{\theta}_1 + D_2\end{aligned}$$

where

- D_{ii}: the inertia to joint i;
- D_{ij}: the coupling of inertia between joint i and j;
- D_{ijj}: the coefficients of the centripetal force to joint i because of the velocity of joint j;
- $D_{iik}(D_{iki})$: the coefficients of the Coriolis force to joint i effected by the velocities of joint i and k;
- D_i: the gravity of joint i.

General dynamic equations of a manipulator

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

$M(\Theta)$: the position dependent $n \times n$-mass matrix of a manipulator
For the example given above:

$$M(\Theta) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

$V(\Theta, \dot{\Theta})$: an $n \times 1$-vector of centripetal and coriolis coefficients
For the example given above:

$$V(\Theta, \dot{\Theta}) = \begin{bmatrix} D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_2\dot{\theta}_1 \\ D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_2\dot{\theta}_1 \end{bmatrix}$$

General dynamic equations of a manipulator (cont.)

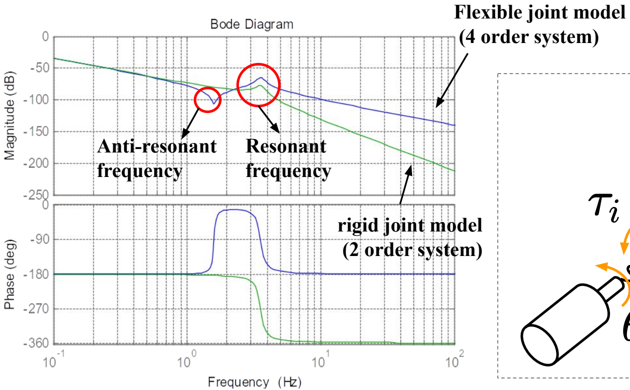
- a term such as $D_{111}\dot{\theta}_1^2$ is caused by coriolis force;
- a term such as $D_{112}\dot{\theta}_1\dot{\theta}_2$ is caused by coriolis force and depends on the (math.) product of the two velocities.
- $G(\Theta)$: a term of velocity, depends on Θ.
  - for the example given above

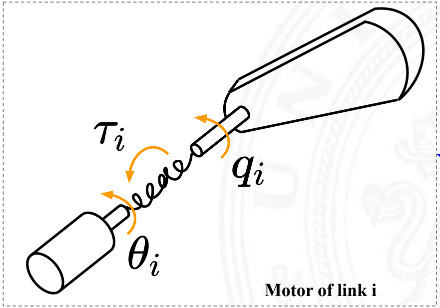
$$G(\Theta) = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}$$

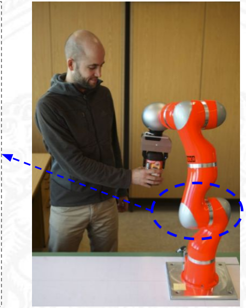
Robot dynamics with flexible joint model

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + DK^{-1}\dot{\tau} + \tau_{ext}$$
$$B\ddot{\theta} + \tau + DK^{-1}\dot{\tau} = \tau_m - \tau_f$$
$$\tau = K(\theta - q)$$

- flexible joint as a two-mass model

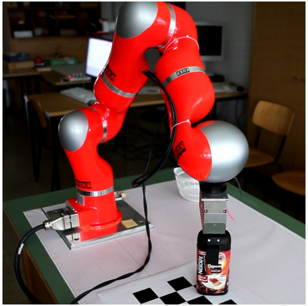






KUKA LWR's model-based control

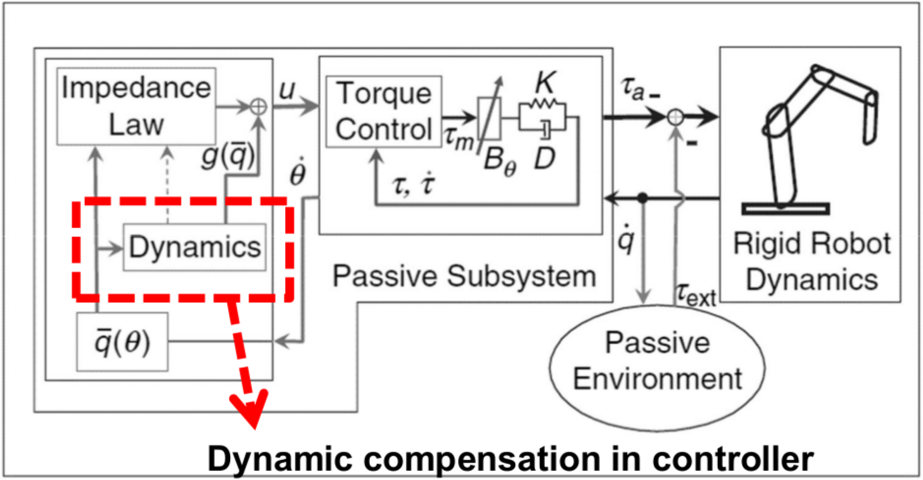
- shortening the motion time without generating overshoots
- giving large reduction of the tracking error



Applications of robot dynamics (cont.)

KUKA iiwa's hand teaching

- Free movement by hand with dynamics compensation on each joint



[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.

Bibliography (cont.)

[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] T. Brogårdh, "Robot control overview: An industrial perspective," *Modeling, Identification and Control*, vol. 30, no. 3, p. 167, 2009.

[13] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*. PhD thesis, Bielefeld University, 2001.

[14] 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.

[15] 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.

[16] J. Canny, *The Complexity of Robot Motion Planning*. MIT press, 1988.

[17] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*. Cambridge, MA, USA: MIT Press, 1992.

[18] O. Khatib, "The Potential Field Approach and Operational Space Formulation in Robot Control," in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.

[19] 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.

[20] R. Geraerts and M. H. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.

[21] 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.

[22] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.

[23] M. J. Mataric, "Interaction and intelligent behavior.," tech. rep., DTIC Document, 1994.

Bibliography (cont.)

[24] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning.," in *AAAI*, vol. 87, pp. 677–682, 1987.

[25] J. Zhang and A. Knoll, *Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control*, pp. 367–385.
Heidelberg: Physica-Verlag HD, 2001.

[26] 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.

[27] A. Meystel, "Nested hierarchical control," 1993.

[28] 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.

[29] 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.

Bibliography (cont.)

[30] 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.

[31] 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.

[32] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration.*
Master thesis, Universität Hamburg, 2015.

[33] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual.*
Addison-Wesley Pub. Co., 1986.

[34] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter.*
Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.

[35] R. Schilling, *Fundamentals of robotics: analysis and control.*
Prentice Hall, 1990.

[36] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, MA, USA: MIT Press, 1990.

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