Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN Faculty
Department of Informatics

Introduction to Robotics

Lecture 9

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 11, 2021

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: gravity, 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
- 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.

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

$V(\Theta, \dot{\Theta})$: an $n \times 1$-vector of centrifugal and Coriolis coefficients

$G(\Theta)$: an $n \times 1$-vector of gravity terms

Inclusion of Nonrigid Body Effects

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

- $F(\Theta, \dot{\Theta})$ is the friction term
- Viscous friction and Coulomb friction
- friction also displays a dependence on the joint position.

How about soft robots? (stretching and bending)

- **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 energy 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:

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

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = F_i, \quad i = 1, \ldots, n$$

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)

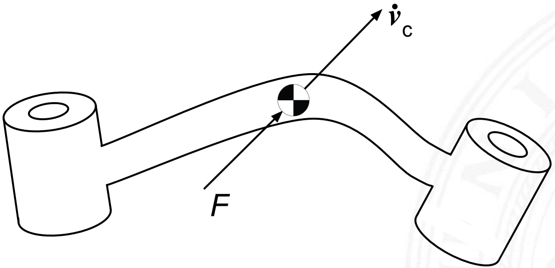
Recursive Newton-Euler Method

- 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

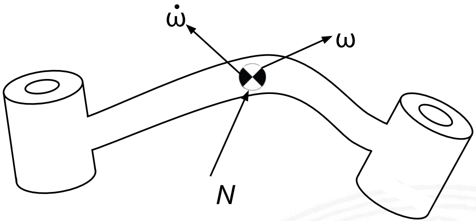
1. Newton's equation

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

where F is the force acting at the center of mass of a body, m is the total mass of the body, $\dot{v}_c$ is the acceleration.



Recursive Newton-Euler Method (cont.)



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

Recursive Newton-Euler Method (cont.)

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

where the scalar elements are given by
mass moments of inertia:

$$
I_{xx} = \int\int\int_V (y^2 + z^2)\rho dv
$$

$$
I_{yy} = \int\int\int_V (x^2 + z^2)\rho dv
$$

$$
I_{zz} = \int\int\int_V (x^2 + y^2)\rho dv
$$

mass products of inertia:

$$
I_{xy} = \int\int\int_V xy\rho dv \; I_{xz} = \int\int\int_V xz\rho dv I_{yz} = \int\int\int_V yz\rho dv
$$

ρ is the material of density, dv is differential volume element

Formulation of robot dynamics

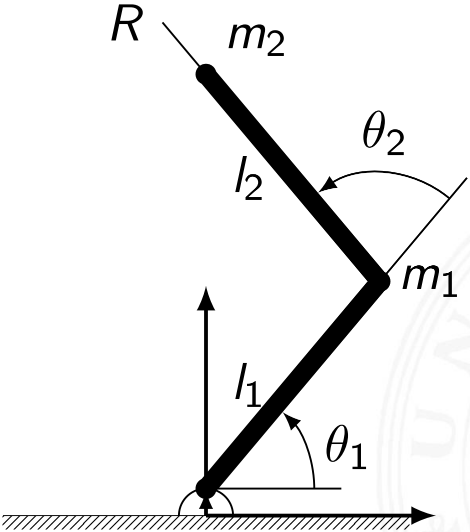
- Combining the different influence factors in the robot specific motion equation from kinematics $(\Theta, \dot{\Theta}, \ddot{\Theta})$
- Practically the Newton-, Euler- and motion-equation for each joint are combined
- Advantages: numerically efficient, applicable for complex geometry, can be modularized

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.



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

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

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

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

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:

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

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

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

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

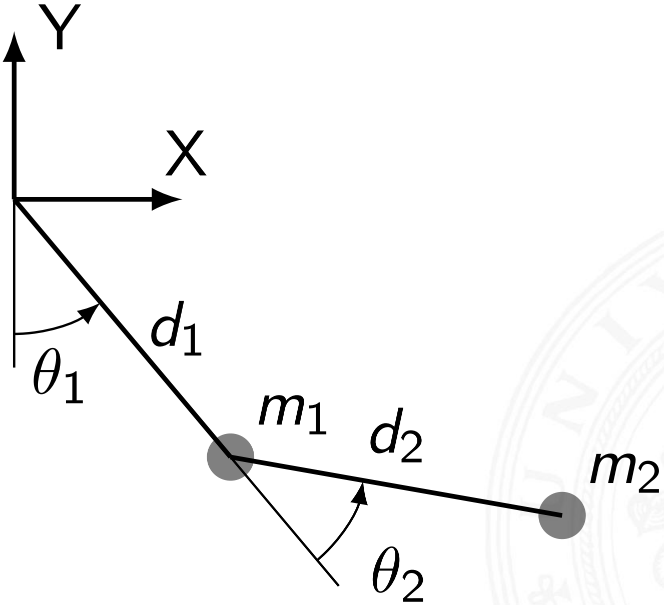
As $\omega_i \times \mathbf{I}_i \omega_i = 0$, the torques at the center of mass of links 1 and 2 are:

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

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

$\mathbf{F}_1, \mathbf{F}_2, \tau_1, \tau_2$ 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:

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

The Lagrangian function is:

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

The force/torque to joint 1 and 2 are:

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

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

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.

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

$M(\Theta)$: 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.)

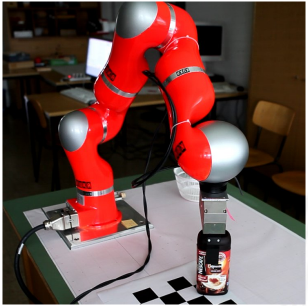
- a term such as $D_{111}\dot{\theta}_1^2$ is caused by coriolis force;
- a term such as $D_{112}\dot{\theta}_1\dot{\theta}_2$ is caused by coriolis force and depends on the (math.) product of the two velocities.
- $G(\Theta)$: a term of velocity, depends on Θ.
  - for the example given above

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

Applications of robot dynamics

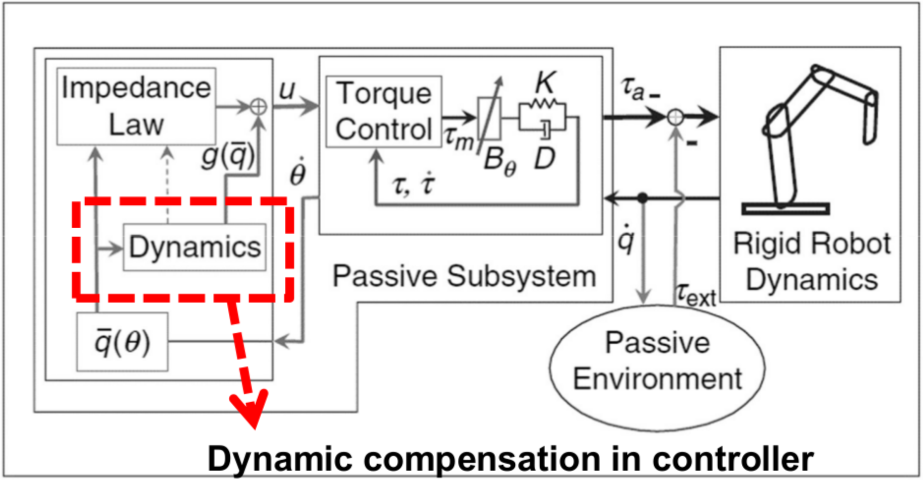
KUKA LWR's model-based control

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



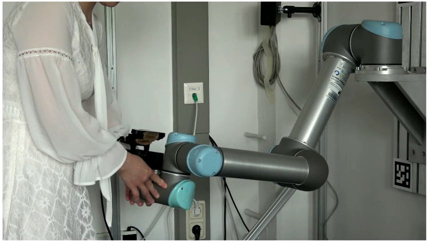
KUKA iiwa's hand teaching

- Free movement by hand with dynamics compensation on each joint

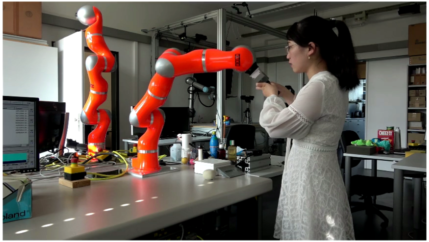


Applications of robot dynamics (cont.)

UR5 hand teaching VS KUKA iiwa's hand teaching



UR5 teach mode



Kuka teach mode

Dynamics libraries

- Rigid Body Dynamics Library (RBDL)
- drake
- frost
- pinocchio

Bibliography

[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.

Bibliography (cont.)

[12] A. Cowley, W. Marshall, B. Cohen, and C. J. Taylor, “Depth space collision detection for motion planning,” 2013.

[13] Hornung, Armin and Wurm, Kai M. and Bennewitz, Maren and Stachniss, Cyrill and Burgard, Wolfram, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, pp. 189–206, 2013.

[14] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, “Manipulation planning on constraint manifolds,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 625–632, 2009.

[15] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

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

[17] L. E. Kavraki, P. Svestka, J. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

Bibliography (cont.)

[18] J. Kuffner and S. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning.," vol. 2, pp. 995–1001, 01 2000.

[19] J. Starek, J. Gómez, E. Schmerling, L. Janson, L. Moreno, and M. Pavone, "An asymptotically-optimal sampling-based algorithm for bi-directional motion planning," *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2015, 07 2015.

[20] D. Hsu, J. . Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proceedings of International Conference on Robotics and Automation*, vol. 3, pp. 2719–2726 vol.3, 1997.

[21] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2118–2124, IEEE, 2019.

[22] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *in Proc. Robotics: Science and Systems*, 2013.

Bibliography (cont.)

[23] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[24] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[25] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477, 2011.

[26] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach.," in *Robotics: Science and Systems*, pp. 1–6, 2016.

[27] J. Ferrer-Mestres, G. Francès, and H. Geffner, "Combined task and motion planning as classical ai planning," *arXiv preprint arXiv:1706.06927*, 2017.

[28] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "MoveIt! Task Constructor for Task-Level Motion Planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Bibliography (cont.)

[29] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[30] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[31] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, IEEE, 2018.

[32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

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

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

Bibliography (cont.)

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

[36] 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.

[37] 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.

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

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

Bibliography (cont.)

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

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

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

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