Introduction to Robotics

Lecture 6

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

May 14, 2021

Outline

Outline (cont.)

Trajectory generation – Recapitulation

The trajectory of a robot with n degrees of freedom (DoF) is a vector of n parametric functions with a common parameter:

Time

$$q(t) = [q^1(t), q^2(t), ..., q^n(t)]^T$$

Trajectory generation – Recapitulation (cont.)

- Deriving a trajectory yields
  - velocity $\dot{q}$
  - acceleration $\ddot{q}$
  - jerk $\dddot{q}$
- Jerk represents the change of acceleration over time, allowing for non-constant accelerations.
- A trajectory is C^k-continuous, if the first k derivatives of its path exist and are continuous.
- A trajectory is defined as *smooth* if it is at least C^2-continuous.

Trajectory generation – Recapitulation (cont.)

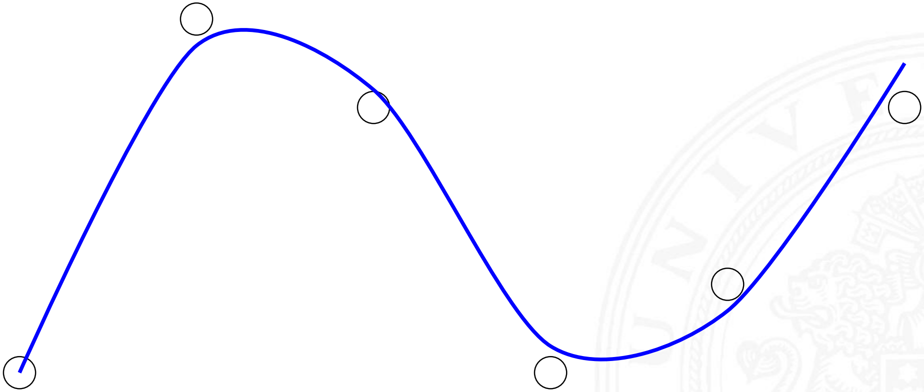
Trajectory generation

- Cartesian space
  - closer to the problem
  - better suited for collision avoidance
- Joint space
  - trajectories are immediately executable
  - limited to direct kinematics
  - allows accounting for joint angle limitations

Trajectory generation – Recapitulation (cont.)

- Linear interpolation
  - respect the minimum velocity constraint
- Trapezoidal interpolation
  - normalization
- Polynomial interpolation.
  - differentiable acceleration
  - cubic polynomials

- Approximation of the relation between x and y (curve, plane, hyperplane) with a different function, given a limited number n of data points $D = \{\mathbf{x}_i, y_i\}$



Approximation

Definition

An approximation is a non-exact representation of something that is difficult to determine precisely (e.g. functions).

Necessary if

- equations are hard to solve
- mathematically too difficult or computationally too expensive

Advantages are

- simple to derive
- simple to integrate
- simple to compute

Approximation (cont.)

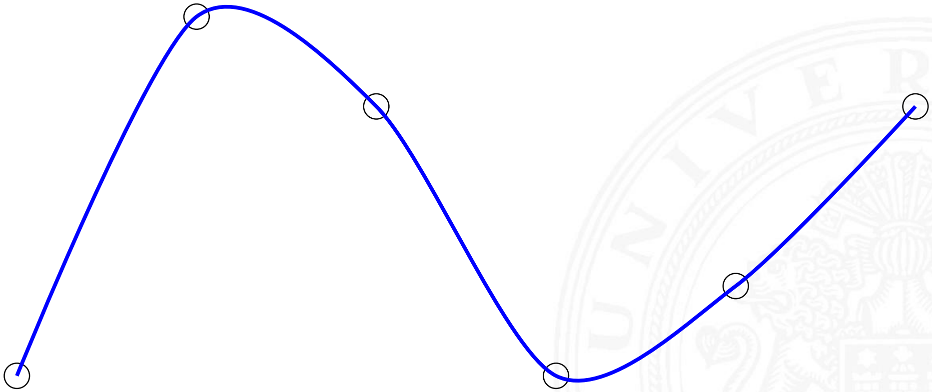
Stone-Weierstrass theorem (1937)

Theorem

- Every non-periodic continuous function on a closed interval can be approximated as closely as desired using algebraic polynomials.
- Every periodic continuous function can be approximated as closely as desired using trigonometric polynomials.

Interpolation

- A special case of approximation is interpolation, where the model exactly matches all data points.
  If many points are given or measurement data is affected by noise, approximation should preferably be used.



Interpolation

Definition

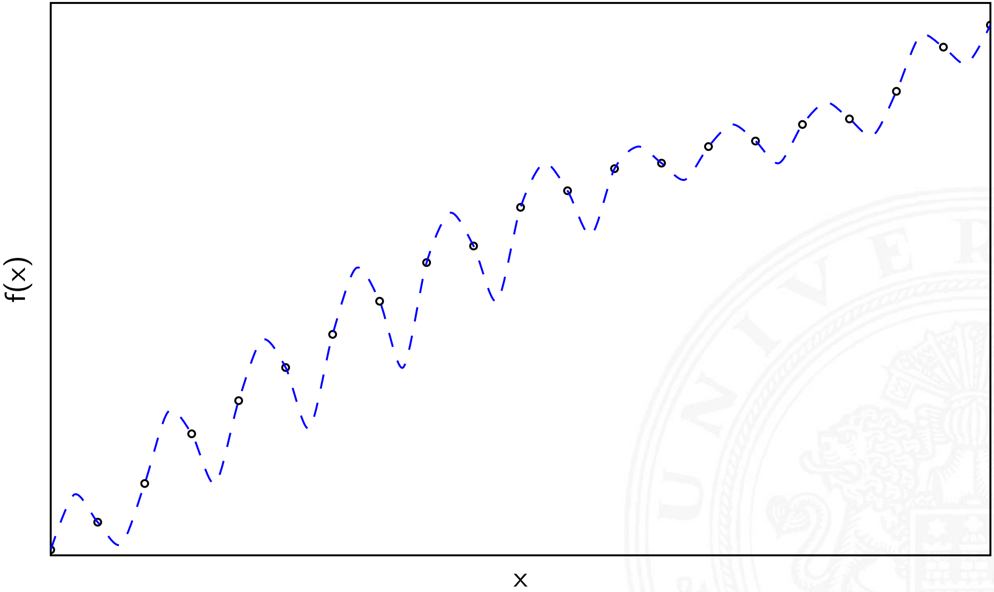
Interpolation is the process of constructing new data points within the range of a discrete set of known data points.

- Interpolation is a kind of approximation.
- A function is designed to match the known data points exactly, while estimating the unknown data in between in an useful way.
- In robotics, interpolation is common for computing trajectories and motion/-controllers.

Interpolation vs. Approximation

- Approximation: Fitting a curve to given data points.
  - Online tool: `https://mycurvefit.com/`
- Interpolation: Defining a curve exactly through all given data points
  - In the case of many, especially noisy, data points, approximation is often better suited than interpolation

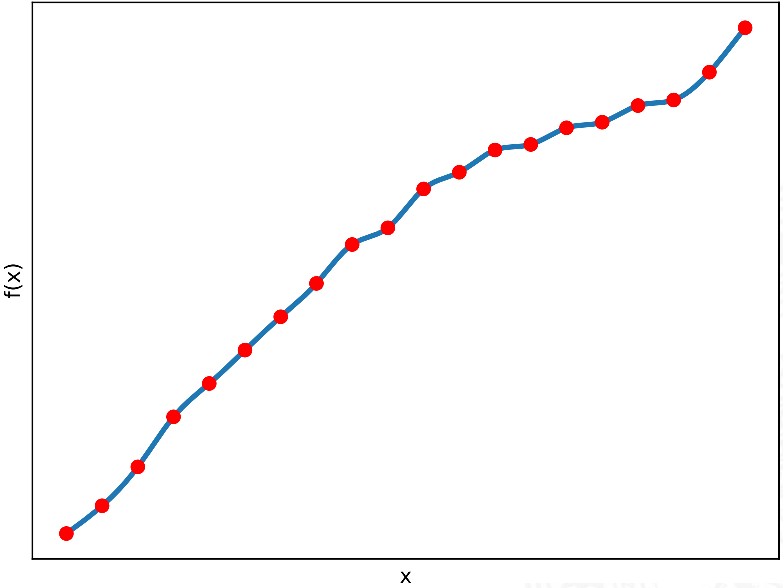
Interpolation with Overfitting



Overfitting example

Complete the sequence: $1, 3, 5, 7, \mathbf{?}$

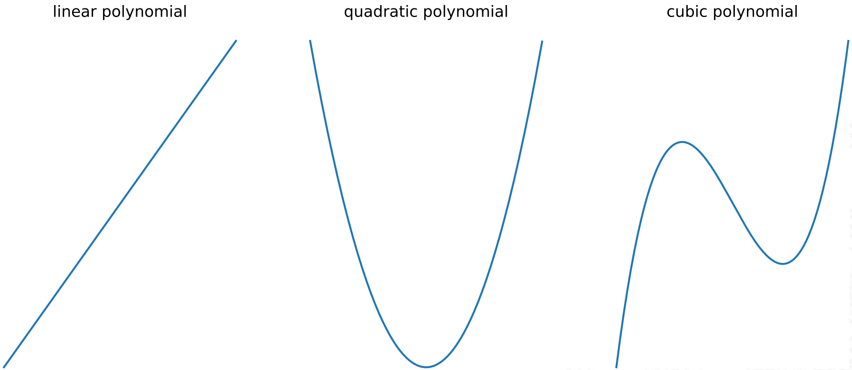
Interpolation without Overfitting



Interpolation basics

- Base
  - subset of a vector space
  - able to represent arbitrary vectors in space
    - finite linear combination
- Uniqueness
  - n^{th}-degree polynomials only have n zero-points
  - resulting system of equations is unique
- Oscillation
  - high-degree polynomials may oscillate due to many extrema
  - workaround: composition of sub-polynomials

Polynomial examples



Whatever the degree n of the polynomial is, there's $n - 1$ turning points.

Interpolation methods

Generation of robot-trajectories in joint-space over multiple stopovers requires appropriate interpolation methods.

Some interpolation methods using polynomials:

- Bernstein-polynomials (Bézier curves)
- Basis-Splines (B-Splines)
- Lagrange-polynomials
- Newton-polynomials

Examples of polynomials interpolation:

- http://polynomialregression.drque.net/online.php
- https://arachnoid.com/polysolve/
- http://www.hvks.com/Numerical/webinterpolation.html

Bernstein-Polynomials

Bernstein-polynomials (named after Sergei Natanovich Bernstein) are real polynomials with integer coefficients.

Definition

Bernstein basis polynomials of degree k are defined as:

$$B_{i,k}(t) = \binom{k}{i}(1-t)^{k-i}t^i, \quad i = 0, 1, \dots, k$$

where $\binom{k}{i}$ is the binomial coefficients, $\binom{k}{i} = \frac{i!}{k!(i-k)!}$ and $k \geqslant i \geqslant 0$.

Bernstein-Polynomials

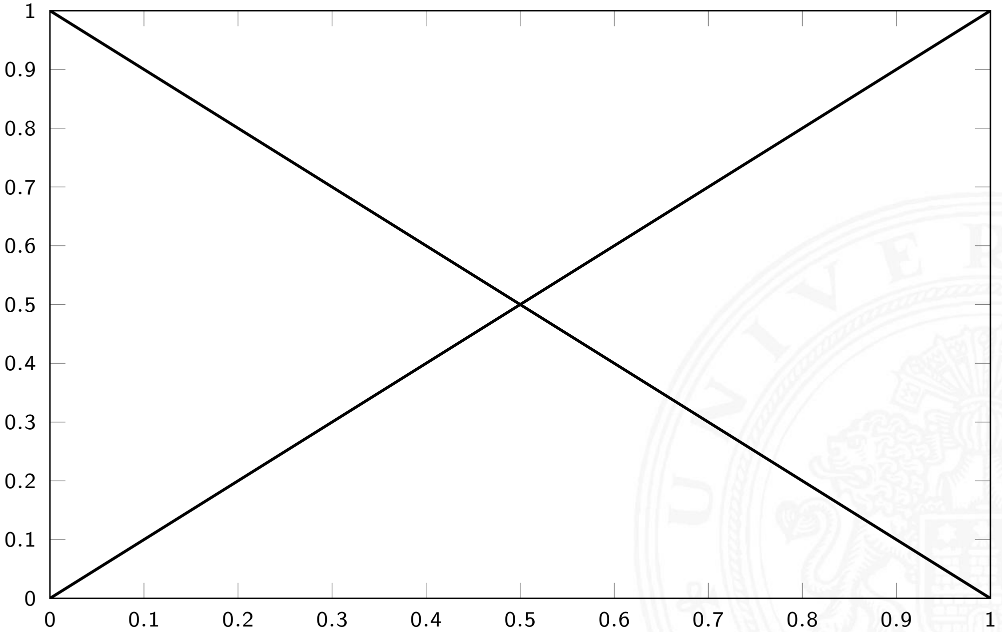
$$B_{i,k}(t) = \binom{k}{i}(1-t)^{k-i}t^i, \quad i = 0, 1, \ldots, k$$

Bernstein Polynomials:

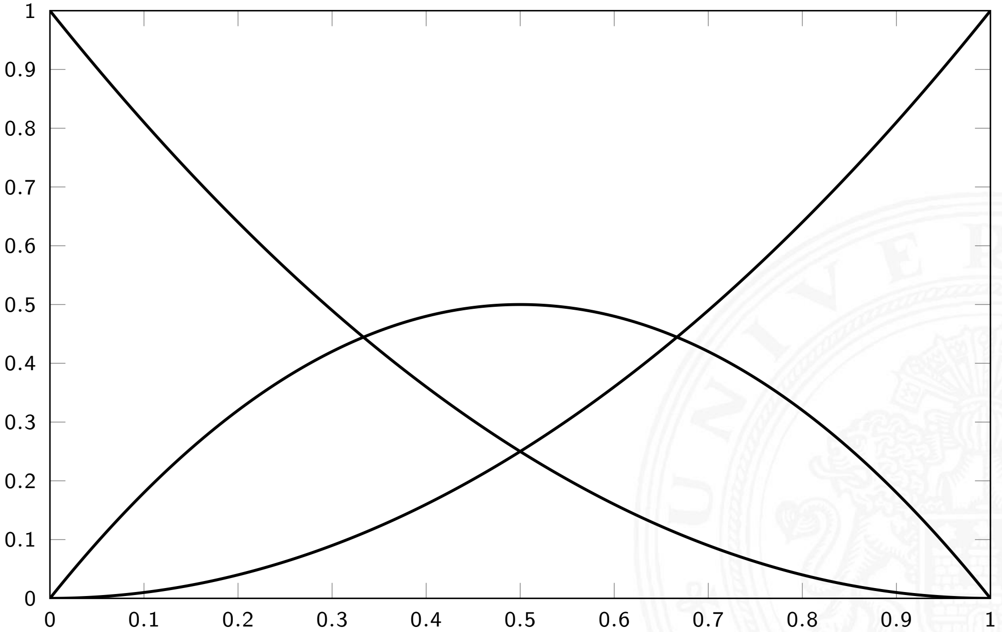
$$\mathbf{y} = \mathbf{b}_0 B_{0,k}(t) + \mathbf{b}_1 B_{1,k}(t) + \cdots + \mathbf{b}_k B_{k,k}(t)$$

where $\mathbf{b}_k$ is Bernstein coefficients.

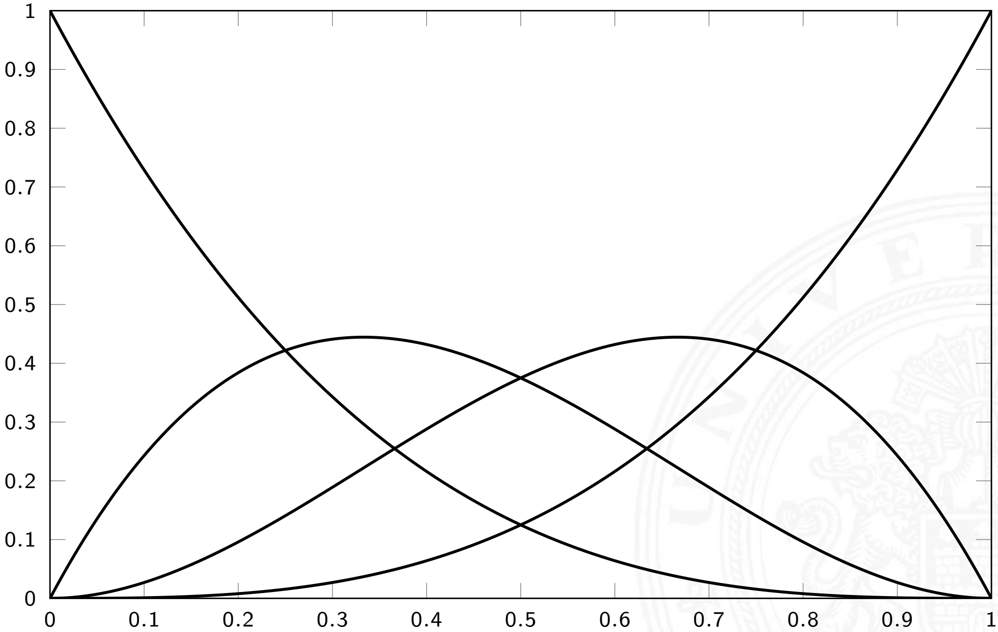
Polynomial of degree 1



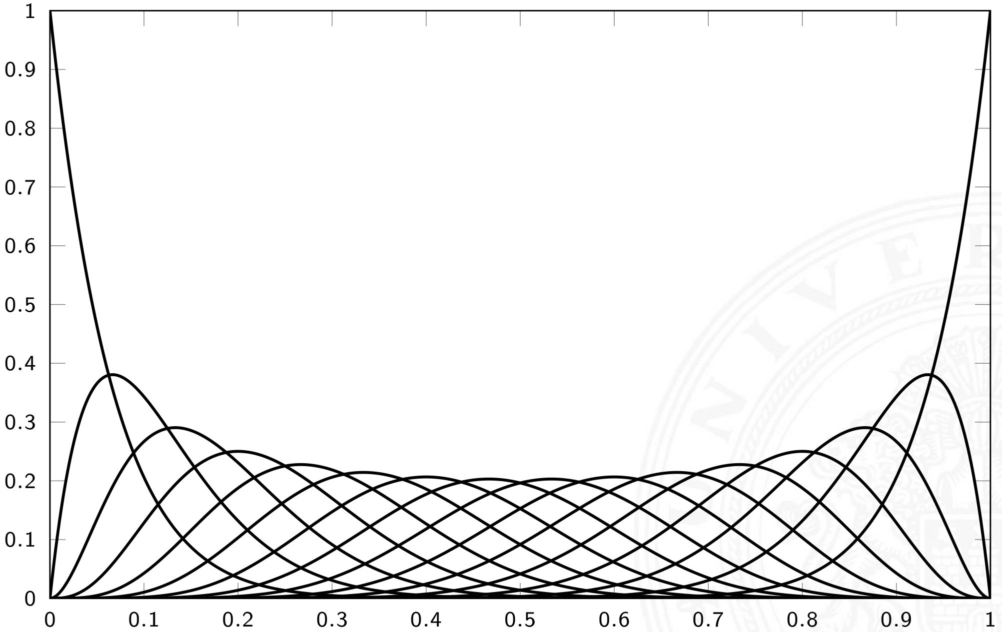
Polynomial of degree 2



Polynomial of degree 3



Polynomial of degree 15



Properties

Properties of Bernstein basis polynomials:

- `base property`: the Bernstein basis polynomials $[B_{i,k} : 0 \leq i \leq k]$ are linearly independent and form a base of the space of polynomials of degree $\leq k$,
- `positivity` $B_{i,k}(t) \geq 0$ for $t \in [0,1]$,
- `decomposition of one`: $\sum_{i=0}^{k} B_{i,k}(t) \equiv \sum_{i=0}^{k} \binom{k}{i} t^i (1-t)^{k-i} \equiv 1$,
- `recursivity`: $B_{i,k-1}(t) = \frac{k-i}{k} B_{i,k}(t) + \frac{i+1}{k} B_{i+1,k}(t)$
- ...

Bézier curves

Bernstein Polynomials:

$$\mathbf{y} = \mathbf{b}_0 B_{0,k}(t) + \mathbf{b}_1 B_{1,k}(t) + \cdots + \mathbf{b}_k B_{k,k}(t)$$

where $\mathbf{b}_k$ is Bernstein coefficients.

If $\mathbf{b}_k$ is a set of control points $P_0, \cdots, P_n$, where n is called its order of the Bézier curve (n $=$ 1 for linear, 2 for quadratic, etc.).

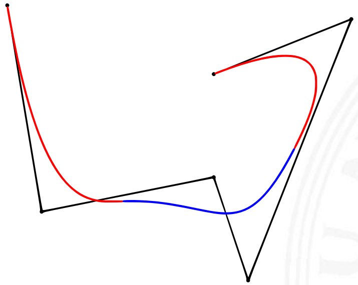
Animation of Bézier curves

Bézier curves for trajectory generation

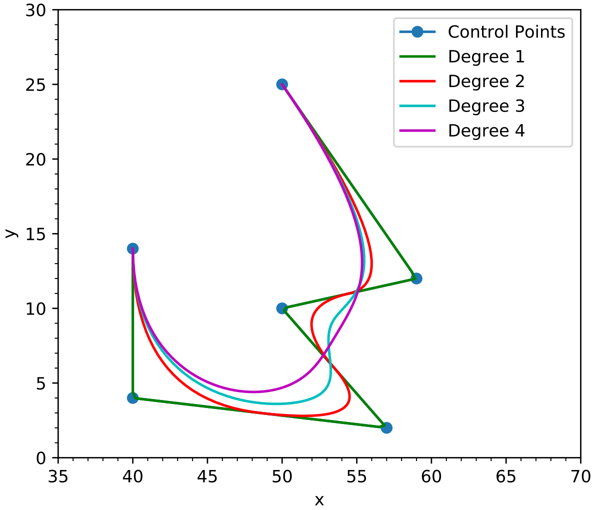
- Cubic polynomials (3^{rd}-degree) most used
- derivatives exist
  - velocity
  - acceleration
  - jerk
- provides smooth trajectory

B-spline curves and basis functions

- A `B-spline` or `basis spline` is a polynomial function that has minimal support with respect to a given degree, smoothness, and domain partition
- A B-spline curve of order k is composed of linear combinations of B-Splines (piecewise) of degree $k-1$ in a set of control points



B-spline curves and basis functions (cont.)



B-spline curves and basis functions (cont.)

Linear splines correspond to piecewise linear functions

Advantages:

- splines are more flexible than polynomials due to their piecewise definition
- still, they are relatively simple and smooth
- prevent strong oscillation
- Generally, 2^{nd} derivatives are continuous at intersections
- also applicable for representing surfaces (CAD modeling)

B-spline curves and basis functions (cont.)

- the domain of B-splines are subdivided by

$$\mathbf{t} = (t_0, t_1, t_2, \ldots, t_m, t_{m+1}, \ldots, t_{m+k}),$$

where

- t: is the **knot vector**, has $m + k$ non-decreasing parameters
- m-th knot span is the half-open inteval $[t_m, t_{m+1})$
- m: is the number of **control points** to be interpolated
- k: is the **order** of the B-spline curve

Definition of B-splines

B-splines $N_{i,k}$ of order k:

- for $k = 1$, the degree is $p = k - 1 = 0$:

$$N_{i,1}(t) = \begin{cases} 1 & : \quad \text{for } t_i \leq t < t_{i+1} \\ 0 & : \quad \text{else} \end{cases}$$

- a recursive definition for $k > 1$

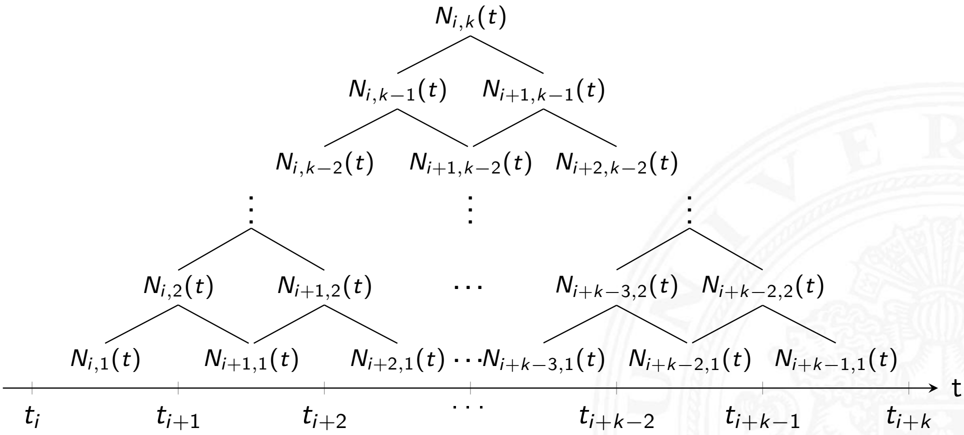
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

with $i = 0, \ldots, m$.

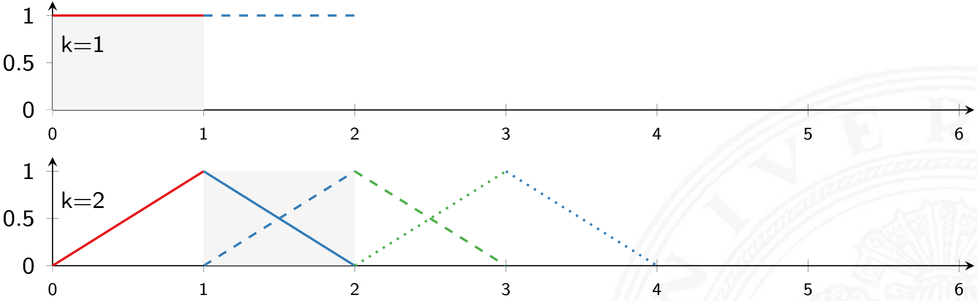
- the above is referred to as the Cox-de Boor recursion formula

Recursive definition of a B-spline

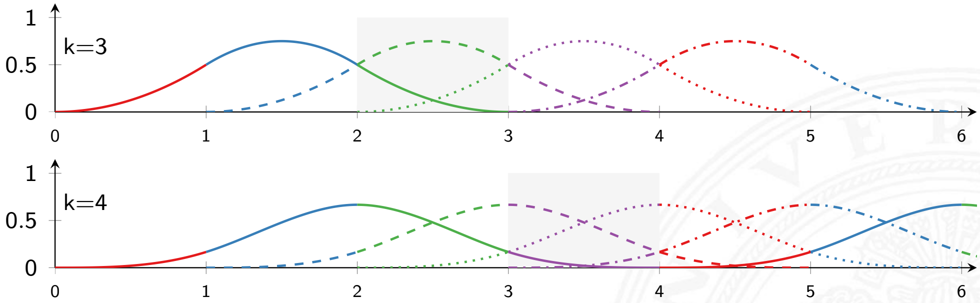
The recursive definition of a B-spline basis function $N_{i,k}(t)$:



Examples of B-splines

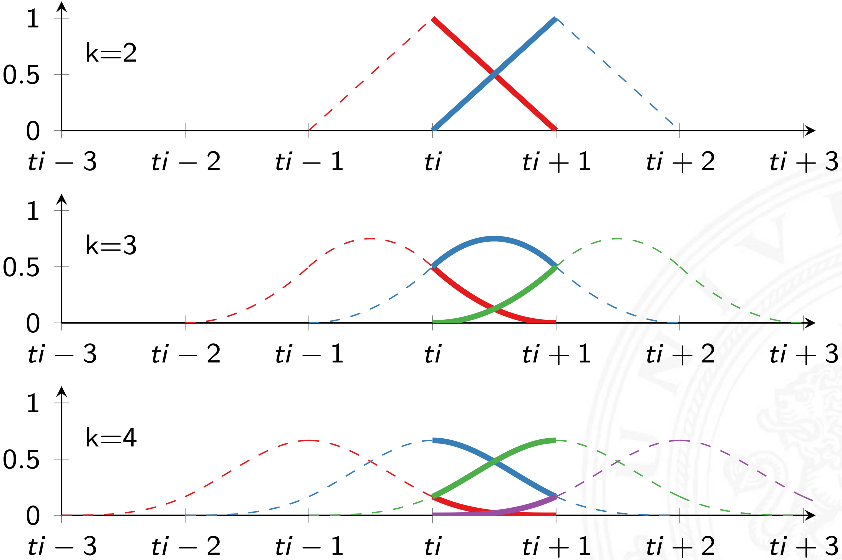


Examples of B-splines



Overlapping

There are $k = p + 1$ overlapping B-splines within an interval.

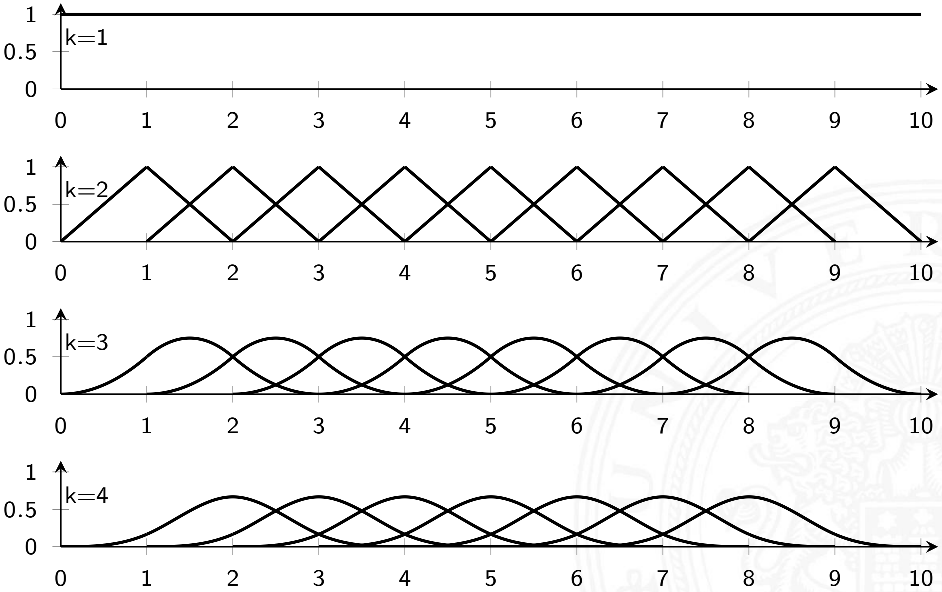


Uniform B-splines

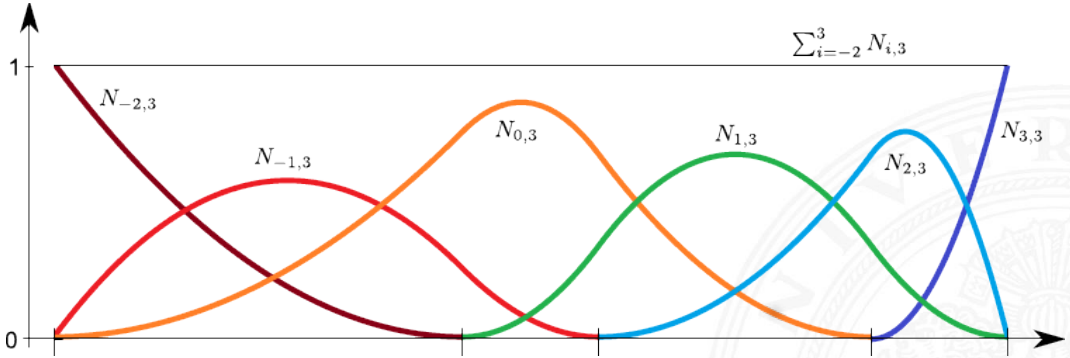
- Distance between uniform B-splines' control points is constant
- Weight-functions of uniform B-splines are periodic
- All functions have the same form
  - Easy to compute

$$B_{m,k} = B_{m+1,k} = B_{m+2,k},$$

Uniform B-splines of order 1 to 4



Non-uniform B-spline of order 3



Properties of B-splines

- Partition of unity: $\sum_{i=0}^{k} N_{i,k}(t) = 1$.
- Positivity: $N_{i,k}(t) \geq 0$.
- Local support: $N_{i,k}(t) = 0$ for $t \notin [t_i, t_{i+k}]$.
- C^{k-2} continuity:
  If the knots $\{t_i\}$ are pairwise different from each other, then

$$N_{i,k}(t) \in C^{k-2}$$

  i.e. $N_{i,k}(t)$ is $(k-2)$ times continuously differentiable.

Construction of a B-spline curve

A B-spline curve can be composed by combining pre-defined control-points with B-spline basis functions:

$$\mathbf{r}(t) = \sum_{j=0}^{m} \mathbf{v}_j \cdot N_{j,k}(t)$$

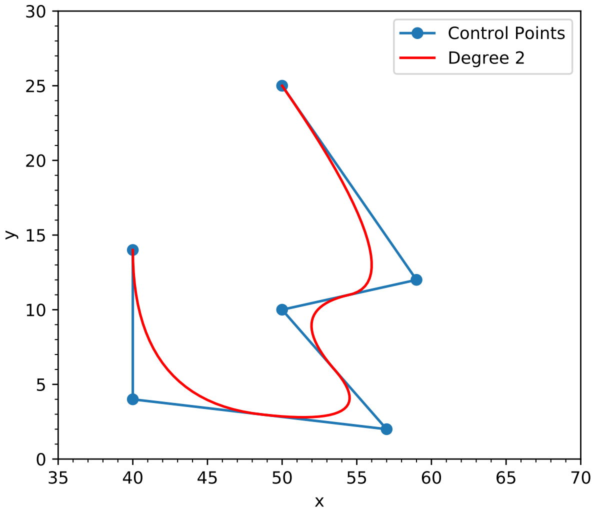
where t is the time, $\mathbf{r}(t)$ is a point on this B-spline curve and $\mathbf{v}_j$ are called its `control points` (`de-Boor points`).

$\mathbf{r}(t)$ is a C^{k-2} continuous curve if the range of t is $[t_{k-1}, t_{m+1}]$.

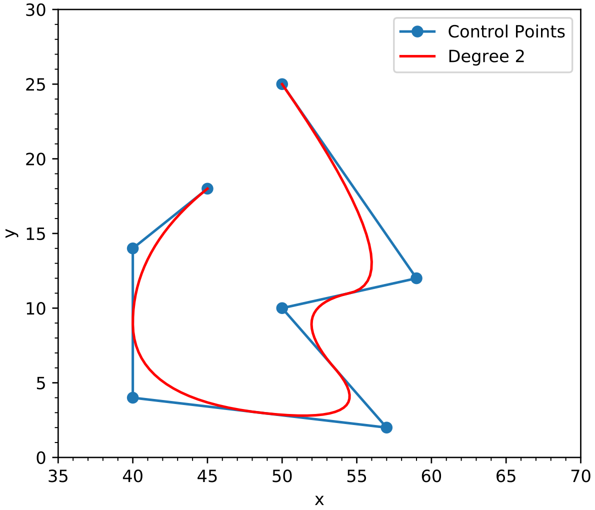
B-Spline curves

- A series of de-Boor points forms a convex hull for the interpolating curve
- Path always constrained to de-Boor point's convex hull
- De-Boor points are of same dimensionality as B-spline curve
- B-spline curves have locality properties
  - control point P_i influences the curve only within the interval $[\tau_i, \tau_{i+p}]$

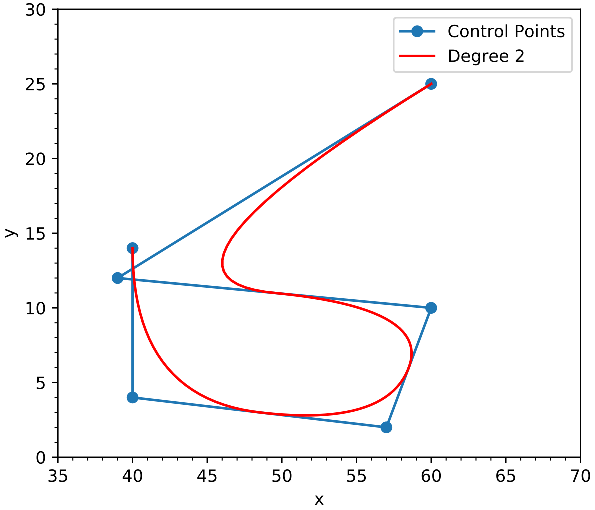
The influence of different control points



The influence of different control points (cont.)



The influence of different control points (cont.)



Generating control points from data points

Question

Given a set of m data points and a degree p, find a B-spline curve of degree p defined by m control points that passes all data points in the given order.

Two methods:

- by solving the following system of equations [9]

$$\mathbf{q}_j(t) = \sum_{j=0}^{m} \mathbf{v}_j \cdot N_{j,k}(t) \Longrightarrow Q = N \cdot V$$

  where $\mathbf{q}_j$ are the data points to be interpolated, $j = 0, \cdots, m$;

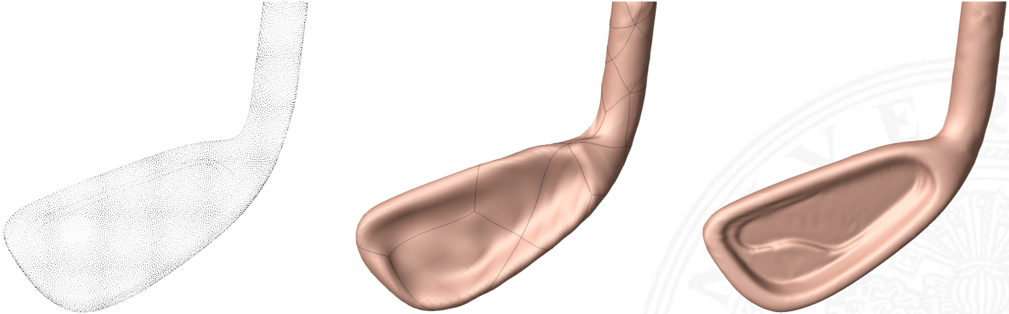
  N is a $m \times m$ matrix;

  V and Q is a $m \times s$ matrices, s is the space dimension.

- by learning, based on gradient-descend.[10]

Surface reconstruction with B-Splines

- Surface reconstruction from laser scan data using B-splines [11]

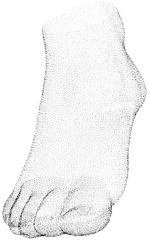


Pointcloud (16,585 points)

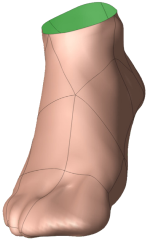
35 patches, 1.36% max. error

285 patches, 0.41% max. error

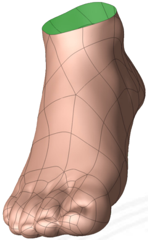
Surface reconstruction with B-Splines (cont.)



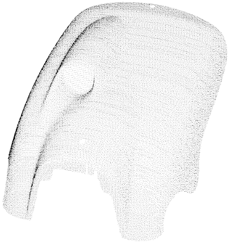
Pointcloud (20,021 points)



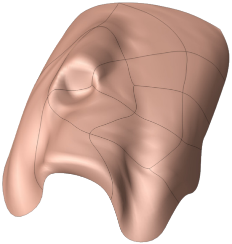
29 patches, 1.20% max. error



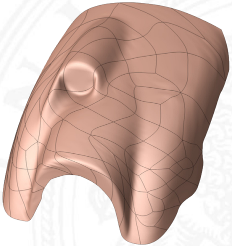
156 patches, 0.27% max. error



Pointcloud (37,974 points)



15 patches, 3.00% max. error

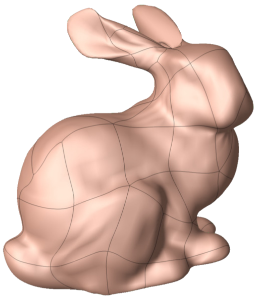


94 patches, 0.69% max. error

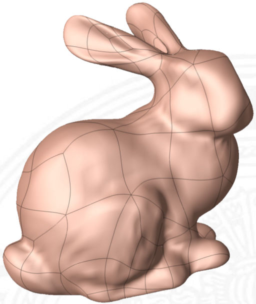
Surface reconstruction with B-Splines (cont.)

- Surface approximation from mesh data (reduced to 30,000 faces)

Mesh (69,473 faces)



72 patches, 4.64% max. error



153 patches, 1.44% max. error

Lagrange interpolation

To match $l+1$ data points (x_i, y_i) $(i = 0, 1, \ldots, l)$ with a polynomial of degree l , the following approach of Lagrange can be used:

$$p_l(x) = \sum_{i=0}^{l} y_i L_i(x)$$

The interpolation polynomial in the Lagrange form is defined as follows:

$$L_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_l)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_l)}$$

$$L_i(x_k) = \begin{cases} 1 \text{ if } i = k \\ 0 \text{ if } i \neq k \end{cases}$$

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.