Introduction to Robotics

Lecture 7

Lasse Einig, Jianwei Zhang
[einig, zhang]@informatik.uni-hamburg.de

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

May 17, 2018

Outline

Outline (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

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.

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

Definition

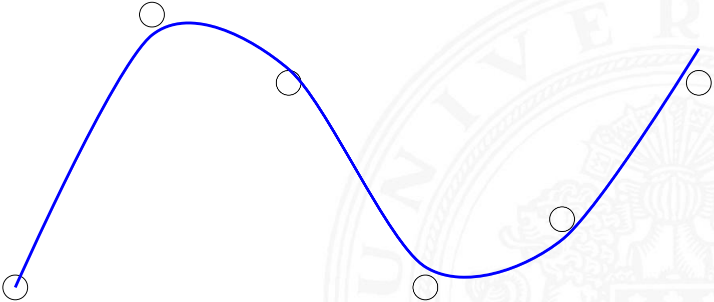
Interpolation is the process of constructing new data points within the range of a discrete set of known data points.

- 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

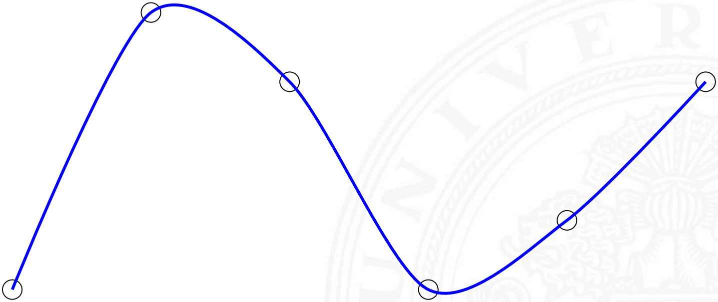
- Approximation: Fitting a curve to given data points.
  - Online tool: `https://mycurvefit.com/`
- Interpolation: Defining a curve exactly through all given data points
  - In the case of many, especially noisy, data points, approximation is often better suited than interpolation

- Approximation of the relation between x and y (curve, plane, hyperplane) with a different function, given a limited number n of data points $D = \{\mathbf{x}_i, y_i\}_{;\, i\in\{1...n\}}$.

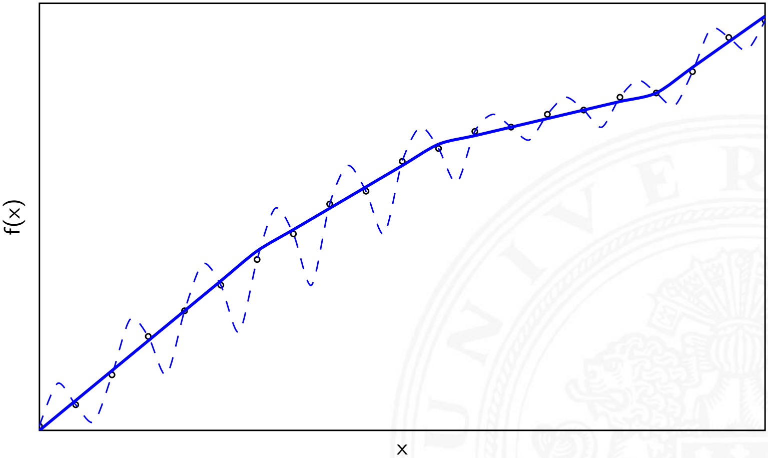


Interpolation vs. Approximation (cont.)

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



Approximation without Overfitting



Overfitting example

Complete the sequence: $1, 3, 5, 7,$ **?**

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
- Oszillation
  - high-degree polynomials may oszillate due to many extrema
  - workaround: composition of sub-polynomials

Interpolation methods

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

Some interpolation methods using polynomials:

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

Examples of polynomials interpolation can be found at

- `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-Polynomials of degree k are defined as:

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

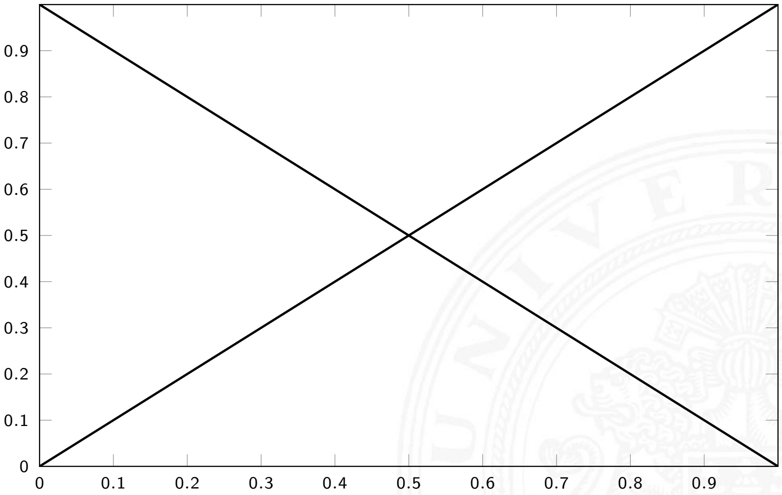
Interpolation with $B_{i,k}$:

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

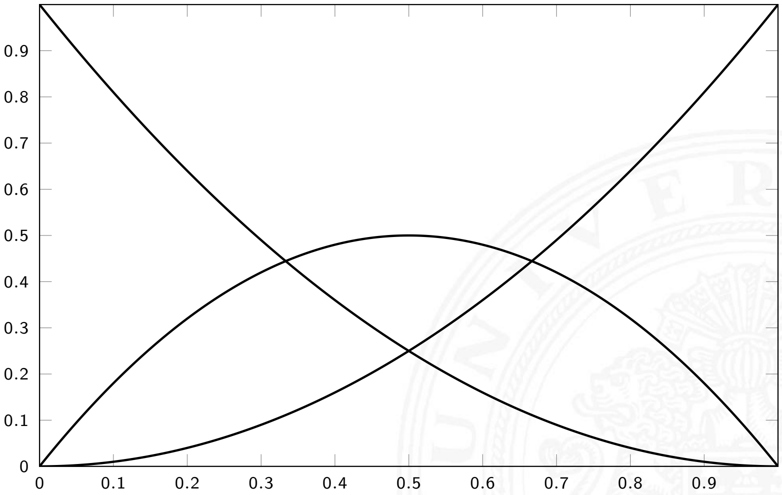
Properties of Bernstein-polynomials:

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

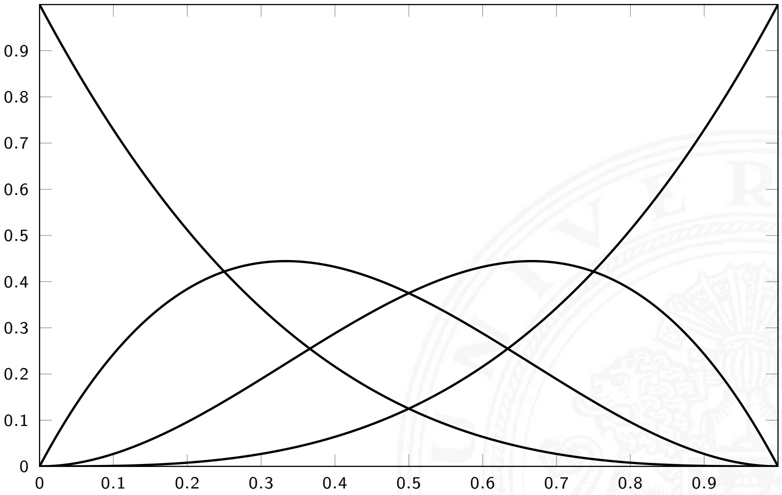
Polynomial of degree 1



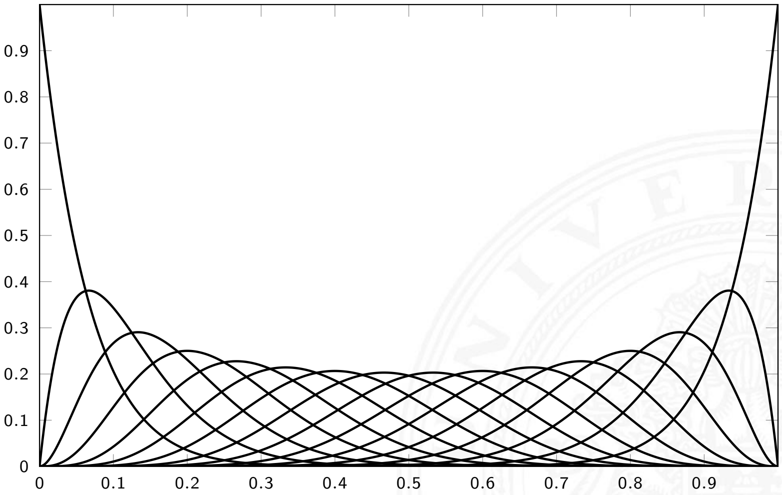
Polynomial of degree 2



Polynomial of degree 3



Polynomial of degree 15



Bernstein polynomials for trajectory generation

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

B-spline curves and basis functions

- Splines are used as `basis` function (hence Basis-Spline)
- B-spline curve is a polynomial
- B-spline curve of order k is composed of B-Splines (piecewise)
- Generally, $k-2$ derivations are continuous at intersections
- B-splines are polynomials based on the following ordered parameters

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

where

- m: is given by the number of points to be interpolated
- k: is the order of the b-spline curve

Definition of B-splines

The following functions are known as normalized B-splines $N_{i,k}$ of order k:

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

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

as well as 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 $i = 0, \ldots, m$.

B-Splines

Linear splines correspond to piecewise linear functions

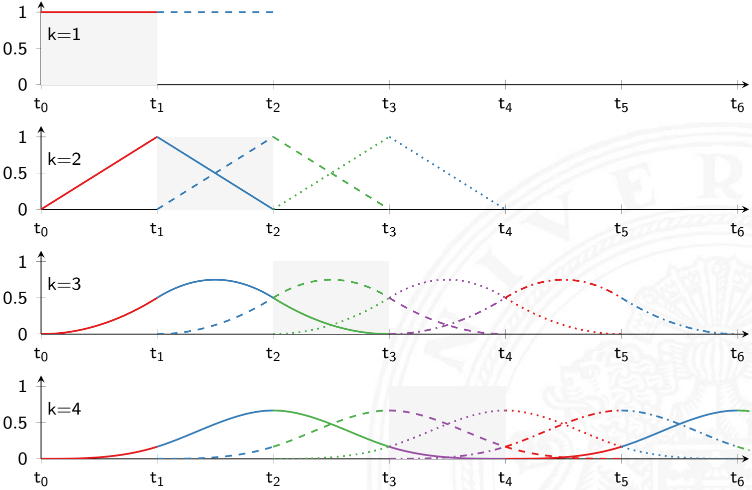
Advantages:

- splines are more flexible than polynomials due to their piecewise definition
- still, they are relatively simple and smooth
- prevent strong oscillation
- the values of the 1^{st} and 2^{nd} derivatives can be defined as constraints
- also applicable for representing surfaces (CAD modeling)

B-Spline curves

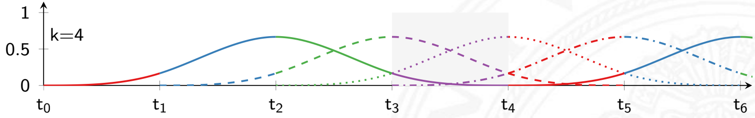
- Path controlled by de-Boor points
- Always constrained to de-Boor point's convex hull
- De-Boor points are of same dimensionality as B-spline curve
- B-spline curves have locality properties similar to Bézier-curves
  - control point P_i influences the curve only within the interval $[\tau_i, \tau_{i+p}]$

Examples of B-splines



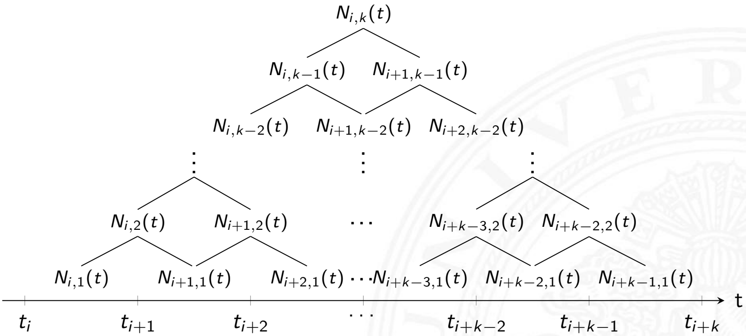
Overlapping

There are $k = p + 1$ overlapping B-splines within an interval.
An example of cubic ($p = 3$) B-splines:

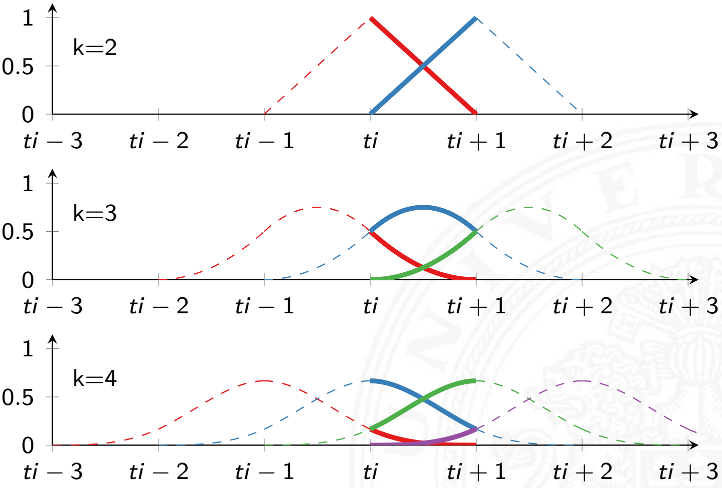


B-Splines of degree n

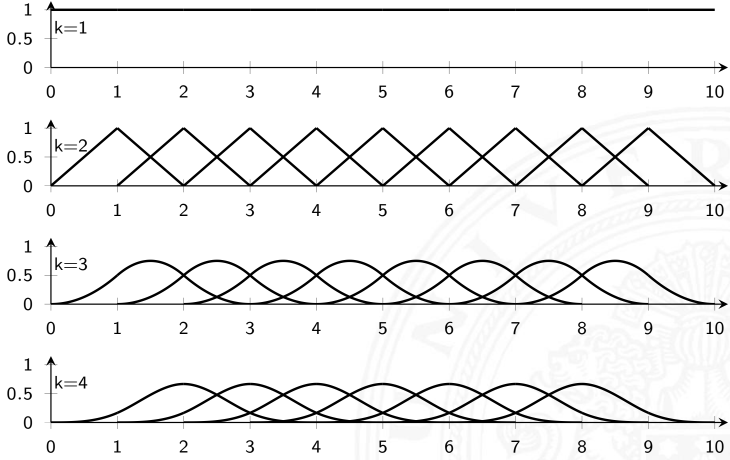
The recursive definition of a B-spline basis function $N_{i,k}(t)$:



B-Splines of degree n in interval $[t_i, t_{i+1}]$



Uniform B-splines of order 1 to 4



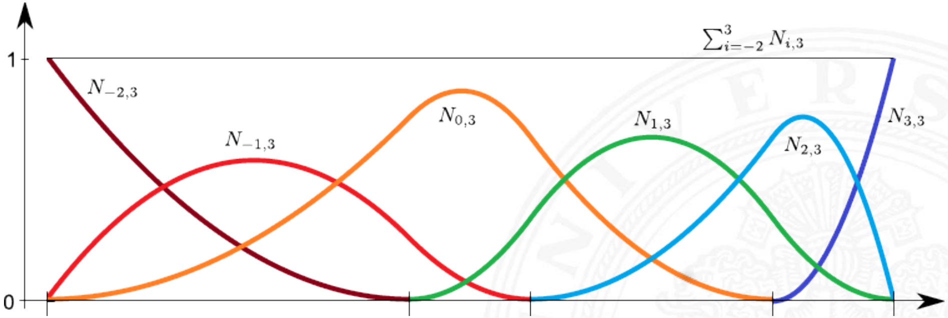
Uniform B-splines

- 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

$$B_{k,d(u)} = B_{k+1,d(u+\Delta u)} = B_{k+2,d(u+2\Delta u)},$$

u represents the control-point's values

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.

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 position, $\mathbf{r}(t)$ 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 $[t_{k-1}, t_{m+1}]$.

Generating control points from data points

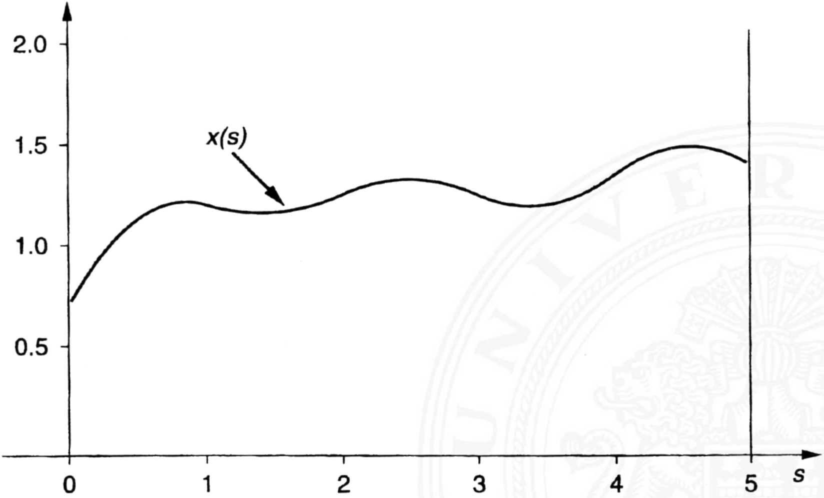
The control points $\mathbf{v}_j$ for interpolation are identical to the data points only if $k = 2$.
A series of control points forms a convex hull for the interpolating curve.
Two methods for generation of control points from data points:

- by solving the following system of equations

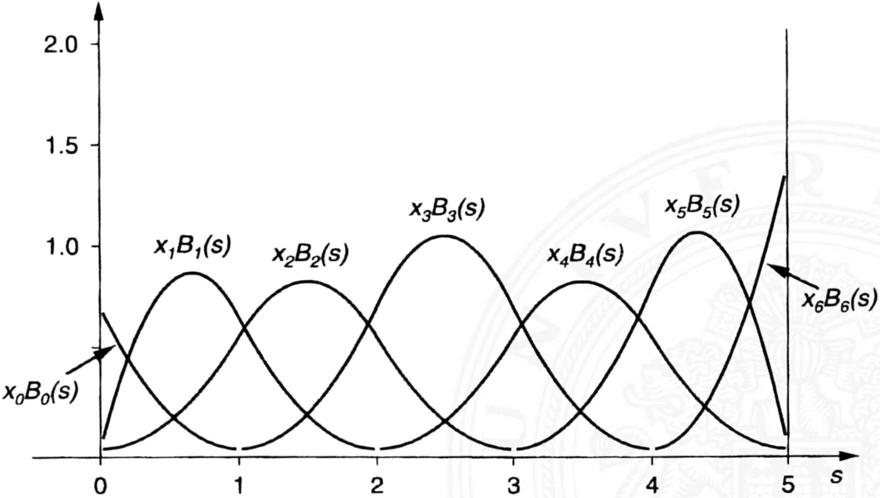
$$\mathbf{q}_j(t) = \sum_{j=0}^{m} \mathbf{v}_j \cdot N_{j,k}(t)$$

  where $\mathbf{q}_j$ are the data points to be interpolated, $j = 0, \cdots, m$.[5]:
- by learning, based on gradient-descend.[6]

Function approximation – 1D example

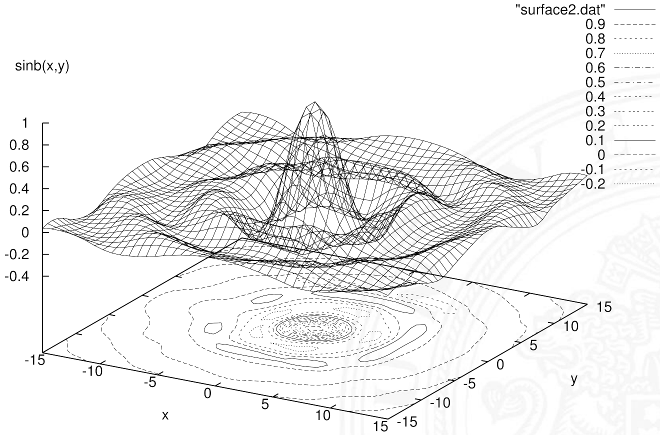


Function approximation – 1D example (cont.)



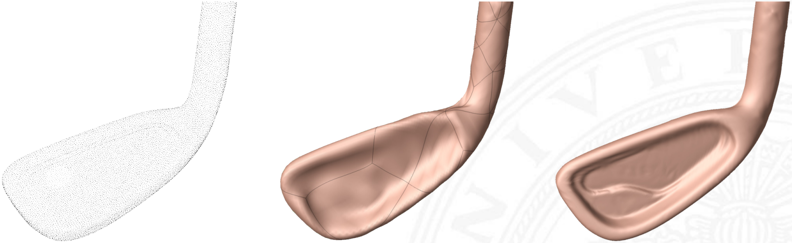
Function approximation – 2D example

Approximation of $sinb(x, y) = sin(\sqrt{x^2 + y^2})/\sqrt{x^2 + y^2}$



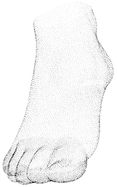
Surface reconstruction with B-Splines

- Surface reconstruction from laser scan data using B-splines [7]

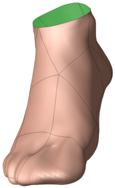


Pointcloud (16,585 points) | 35 patches, 1.36% max. error | 285 patches, 0.41% max. error

Surface reconstruction with B-Splines (cont.)



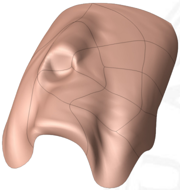
Pointcloud (20,021 points)



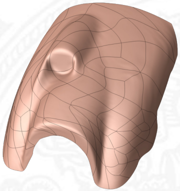
29 patches, 1.20% max. error

156 patches, 0.27% max. error

Pointcloud (37,974 points)



15 patches, 3.00% max. error

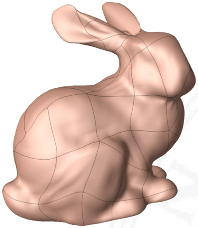


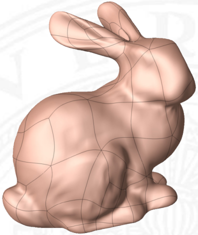
94 patches, 0.69% max. error

Surface reconstruction with B-Splines (cont.)

- Surface reconstruction 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] 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.

[2] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*.
Artificial Intelligence Series, MIT Press, 1981.

[3] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*.
Always learning, Pearson Education, Limited, 2013.

[4] J. F. Engelberger, *Robotics in service*.
MIT Press, 1989.

[5] 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.

Bibliography (cont.)

[6] 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.

[7] 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.

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

[9] 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.

Bibliography (cont.)

[10] 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.

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

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

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

[14] 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.

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

[16] 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.

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

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

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

Bibliography (cont.)

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

[21] 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.

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

[23] 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.

[24] 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.)

[25] 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.

[26] 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.

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

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

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

Bibliography (cont.)

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

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

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