B-Splines for Robotic Applications

Randal W. Beard
Brigham Young University

Updated: March 7, 2024

Abstract

These notes provide an overview of B-splines and their applications to robotics. In particular, we have in mind applications to path planning for aerial vehicles, model predictive control, and fixed-lag state estimation.

1 Motivation

The objective of these notes is to explore spline methods for robotic applications. In general, the position of a robot in Euclidian space can be described by a time parametrized trajectory $\mathbf{p}(t) \in \mathbb{R}^3$, $t \in [a, b]$. The time parametrized trajectory can be parameterized using a weighted sum of basis function as

$$\mathbf{p}(t) = \sum_{m=0}^{N-1} \mathbf{c}_m \phi_m(t),$$

where $\mathbf{c}_m \in \mathbb{R}^n$, and $\phi_m(t)$ are a set of basis functions. For example, the basis functions could be the set of polynomial power function $\phi_m(t) = t^m/m|!$, or the set of sinusoidal function $\phi_m(t) = \sin(\frac{2\pi m}{N} t)$. The disadvantage of both the polynomial power functions and sinusoidal functions is that the basis functions are defined for all $t \in [a, b]$ and so each control points $\mathbf{c}_m$ influences the entire trajectory. Another disadvantage is that a large number of basis functions may be required to represent complicated trajectories.

In these notes, we will use B-spline basis functions which have a number of very nice properties that we will explore. In particular, a *B-spline* trajectory has the following form

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{t}), \tag{1}$$

where $\mathbf{c}_m \in \mathbb{R}^n$ are the control points, $\mathbf{t} = (\tau_0, \tau_1, \tau_2, \dots, \tau_T)$ are called the knot points where $i < j \implies \tau_i \leq \tau_j$, and $b_m^d(t, \mathbf{t})$ are the B-spline basis functions of degree d, given the knot sequence $\mathbf{t}$. The spline trajectories will be defined for t in the interval, i.e., $t \in [\tau_d, \tau_{d+M}]$, which we will write as $t \in \operatorname{span}(\mathbf{t})$

1.1 B-Spline Basis Functions

The B-spline basis function are defined by the recursive formula:

$$b_m^0(t, \mathbf{t}) = \begin{cases} 1 & \text{if } \tau_m \le t \le \tau_{m+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$b_m^d(t, \mathbf{t}) = w_m^d(t, \mathbf{t}) b_m^{d-1}(t, \mathbf{t}) + \left[1 - w_{m+1}^d(t, \mathbf{t})\right] b_{m+1}^{d-1}(t, \mathbf{t}), \tag{3}$$

$$w_m^d(t, \mathbf{t}) = \begin{cases} \frac{t-\tau_m}{\tau_{m+d}-\tau_m}, & \tau_{m+d} \ne \tau_m \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Zero degree basis

For example, if the knot vector is given by

$$\mathbf{t} = [\tau_0, \tau_1, \tau_2] \circeq [0, 1, 2],$$

then there are two basis function of degree $d = 0$ given by

$$b_0^0(t, \mathbf{t}) = \begin{cases} 1 & \text{if } \tau_0 \le t \le \tau_1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_1^0(t, \mathbf{t}) = \begin{cases} 1 & \text{if } \tau_1 \le t \le \tau_2 \\ 0 & \text{otherwise} \end{cases}$$

where $b_0^1(t, \mathbf{t})$ and $b_1^0(t, \mathbf{t})$ are shown in Figure 1. Note that $b_1^0(t, \mathbf{t}) = b_0^0(t-1, \mathbf{t})$, or in other words, all zero degree basis functions are shifted versions of the central zero degree basis function $b_0^0(t, \mathbf{t})$. Additional zero degree basis function can be defined by expanding the knot vector to $\mathbf{t} = [0, \ldots, M]$, where $b_m^0(t, \mathbf{t}) = b_0^0(t - m; \mathbf{t})$ for $m \le M$.

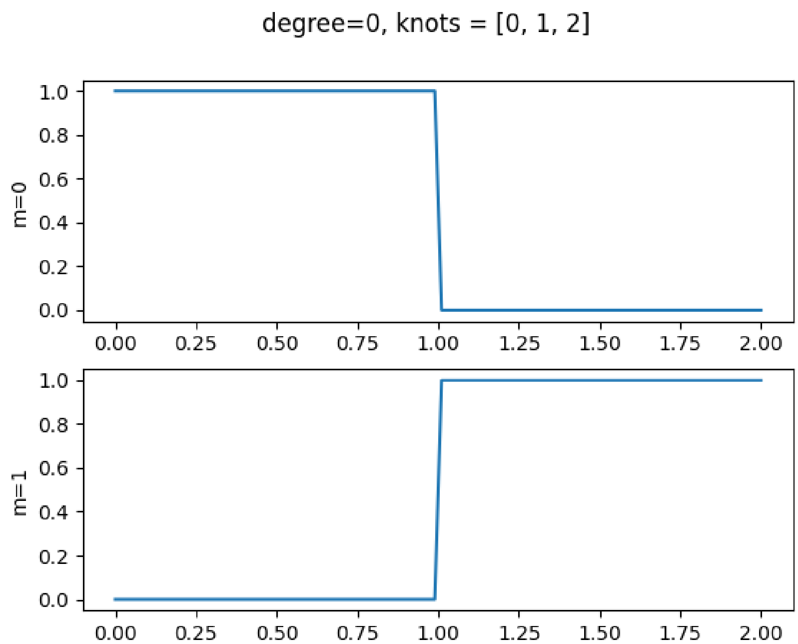

Figure 1: Zeros degree spline basis

First degree basis

If the knot vector is given by

$$\mathbf{t} = [\tau_0, \tau_1, \tau_2, \tau_3, \tau_4] \circeq [0, 0, 1, 2, 2],$$

then there are $2d + 1 = 3$ unique basis function of degree $d = 1$ given by

$$b_0^1(t, \mathbf{t}) = \frac{t - \tau_0}{\tau_1 - \tau_0} b_0^0(t, \mathbf{t}) + \frac{\tau_2 - t}{\tau_2 - \tau_1} b_1^0(t, \mathbf{t}) = \begin{cases} 0 & \text{if } t_0 = 0 \leq t \leq t_1 = 0 \\ 1 - t & \text{if } t_1 = 0 \leq t \leq t_2 = 1 \end{cases}$$

$$b_1^1(t, \mathbf{t}) = \frac{t - \tau_1}{\tau_2 - \tau_1} b_1^0(t, \mathbf{t}) + \frac{\tau_3 - t}{\tau_3 - \tau_2} b_2^0(t, \mathbf{t}) = \begin{cases} t & \text{if } t_1 = 0 \leq t \leq t_2 = 1 \\ 2 - t & t_2 = 1 \leq t \leq t_3 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$b_2^1(t, \mathbf{t}) = \frac{t - \tau_2}{\tau_3 - \tau_2} b_2^0(t, \mathbf{t}) + \frac{\tau_4 - t}{\tau_4 - \tau_3} b_3^0(t, \mathbf{t}) = \begin{cases} t - 1 & \text{if } t_2 = 1 \leq t \leq t_3 = 2 \\ 0 & \text{if } t_3 = 2 \leq t \leq t_4 = 2 \end{cases}$$

where $b_0^1(t, \mathbf{t})$, $b_1^1(t, \mathbf{t})$, and $b_2^1(t, \mathbf{t})$ are shown on the left in Figure 2. If the

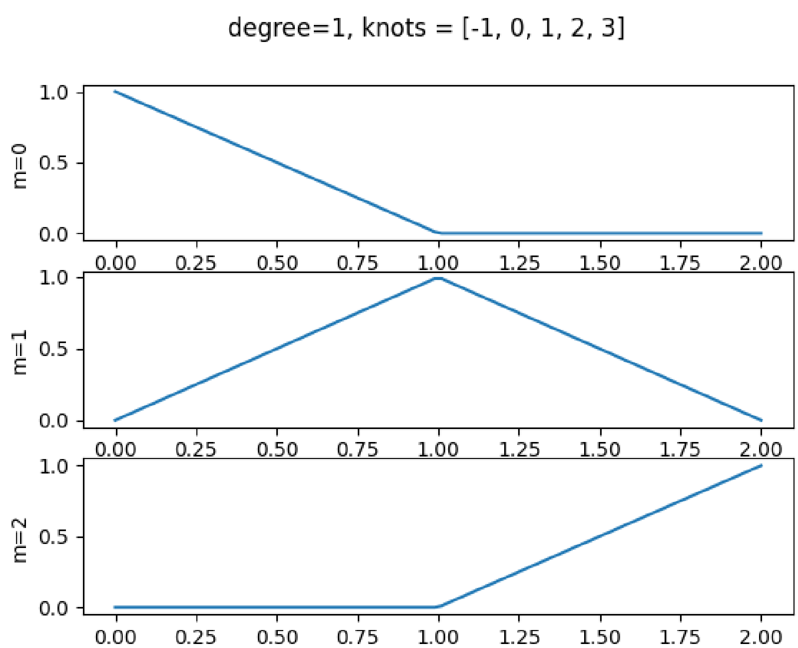

Figure 2: Degree one spline basis

knot vector is expanded by one time unit to

$$\mathbf{t}' = [\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5] \circeq [0, 0, 1, 2, 3, 3],$$

then there are still only three unique basis vectors, but $b_2^1(t, \mathbf{t}')$ is a shifted version $b_1^1(t, \mathbf{t})$ and $b_3^1(t, \mathbf{t}')$ is a shifted version $b_2^1(t, \mathbf{t})$, as shown on the right in Figure 2.

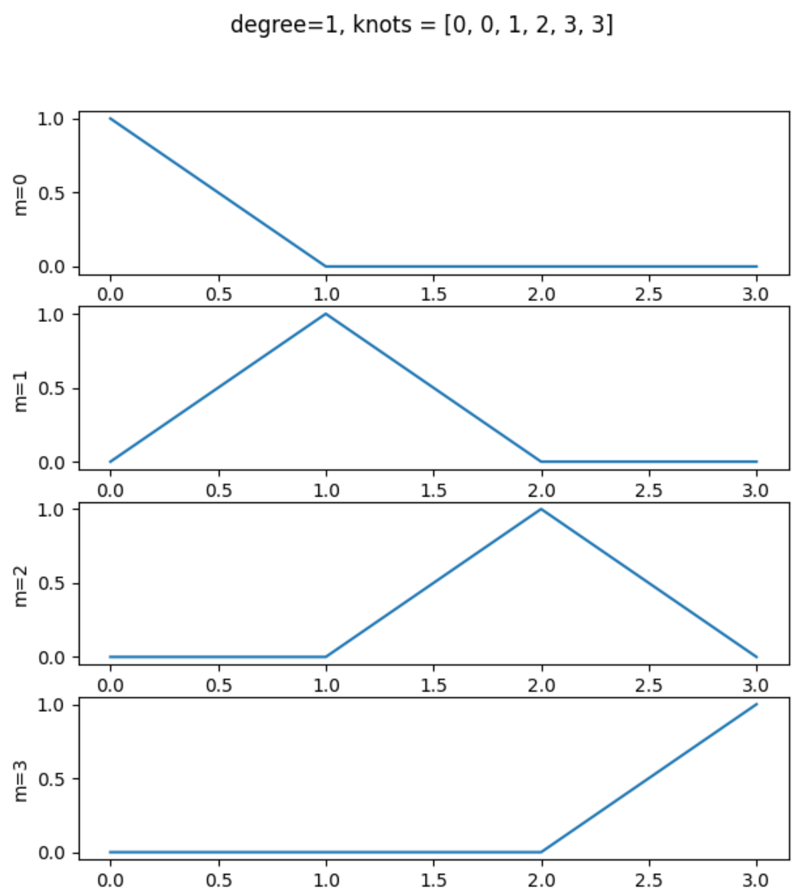

Figure 3: Degree one spline basis

Second degree basis

If the knot vector is given by

$$\mathbf{t} = [\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7, \tau_8] \circeq [0, 0, 0, 1, 2, 3, 3, 3],$$

then there are $2d + 1 = 5$ unique basis function of degree $d = 2$ given by

$$b_0^2(t, \mathbf{t}) = \frac{t - \tau_0}{\tau_2 - \tau_0} b_0^1(t, \mathbf{t}) + \frac{\tau_3 - t}{\tau_3 - \tau_1} b_1^1(t, \mathbf{t}) = \begin{cases} (1-t)^2 & \text{if } 0 \le t \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_1^2(t, \mathbf{t}) = \frac{t - \tau_1}{\tau_3 - \tau_1} b_1^1(t, \mathbf{t}) + \frac{\tau_4 - t}{\tau_4 - \tau_2} b_2^1(t, \mathbf{t}) = \begin{cases} t(2 - \frac{3}{2}t) & \text{if } 0 \le t \le 1 \\ \frac{(2-t)^2}{2} & 1 \le t \le 2 \\ 0 & \text{otherwise} \end{cases}$$

$$b_2^2(t, \mathbf{t}) = \frac{t - \tau_2}{\tau_4 - \tau_2} b_2^1(t, \mathbf{t}) + \frac{\tau_5 - t}{\tau_5 - \tau_3} b_3^1(t, \mathbf{t}) = \begin{cases} \frac{t^2}{2} & \text{if } 0 \le t \le 1 \\ -\frac{3}{2}t^2 + \frac{7}{2}t - \frac{3}{2} & \text{if } 1 \le t \le 2 \\ \frac{(3-t)^2}{2} & \text{if } 2 \le t \le 3 \\ 0 & \text{otherwise} \end{cases}$$

$$b_3^2(t, \mathbf{t}) = \frac{t - \tau_2}{\tau_4 - \tau_2} b_2^1(t, \mathbf{t}) + \frac{\tau_5 - t}{\tau_5 - \tau_3} b_3^1(t, \mathbf{t}) = \begin{cases} \frac{(t-1)^2}{2} & \text{if } 1 \le t \le 2 \\ -\frac{3}{2}t^2 + \frac{15}{2}t - \frac{15}{2} & \text{if } 2 \le t \le 3 \\ 0 & \text{otherwise} \end{cases}$$

$$b_4^2(t, \mathbf{t}) = \frac{t - \tau_2}{\tau_4 - \tau_2} b_2^1(t, \mathbf{t}) + \frac{\tau_5 - t}{\tau_5 - \tau_3} b_3^1(t, \mathbf{t}) = \begin{cases} (t-2)^2 & \text{if } 2 \le t \le 3 \\ 0 & \text{otherwise} \end{cases}.$$

The unique second degree basis function are shown on the left in Figure 5. Expanding the knot vector by one time unit to

$$\mathbf{t}' = [\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5] \circeq [0, 0, 0, 1, 2, 3, 4, 4, 4],$$

still results in $2d + 1$ unique basis vectors, but $b_3^2(t, \mathbf{t}')$ is a right-shifted version of $b_2^2(t, \mathbf{t})$, and $b_4^2(t, \mathbf{t}')$ and $b_5^2(t, \mathbf{t})$ are a right-shifted versions of $b_3^2(t, \mathbf{t}')$ and $b_4^2(t, \mathbf{t})$, as shown on the right in Figure 5.

Similarly, the unique fourth order and ninth order basis functions are shown in Figures 6 and 7 respectively.

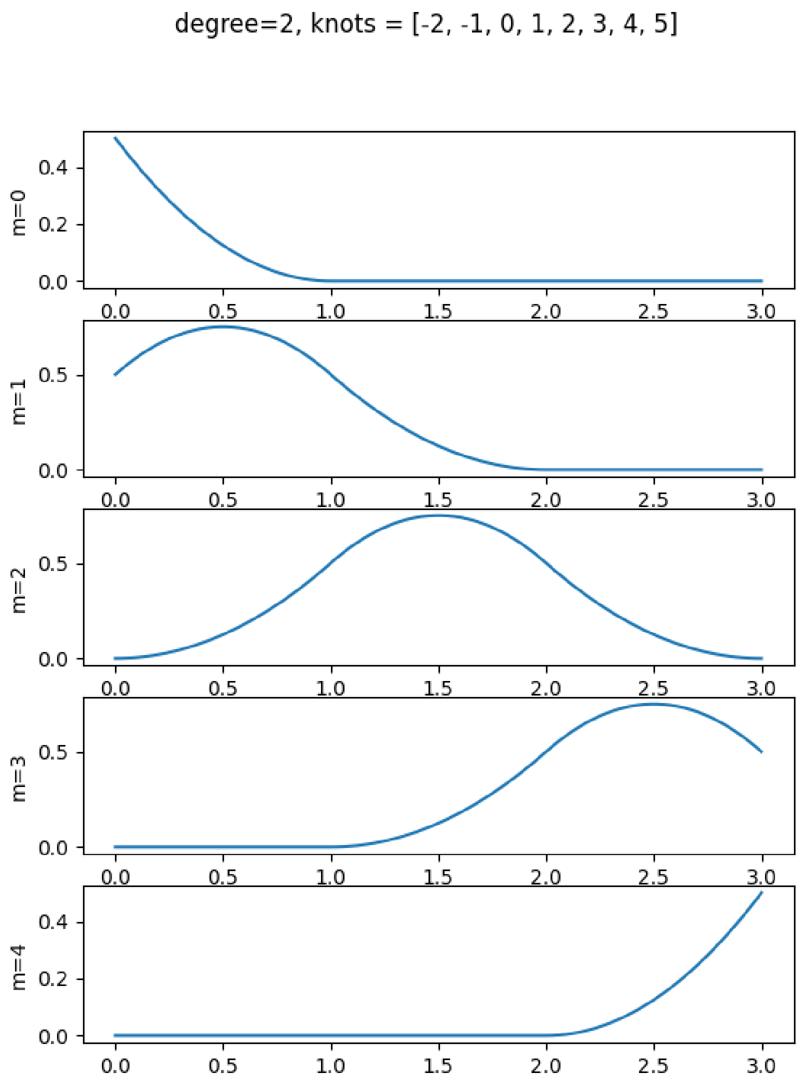

Figure 4: Second degree spline basis.

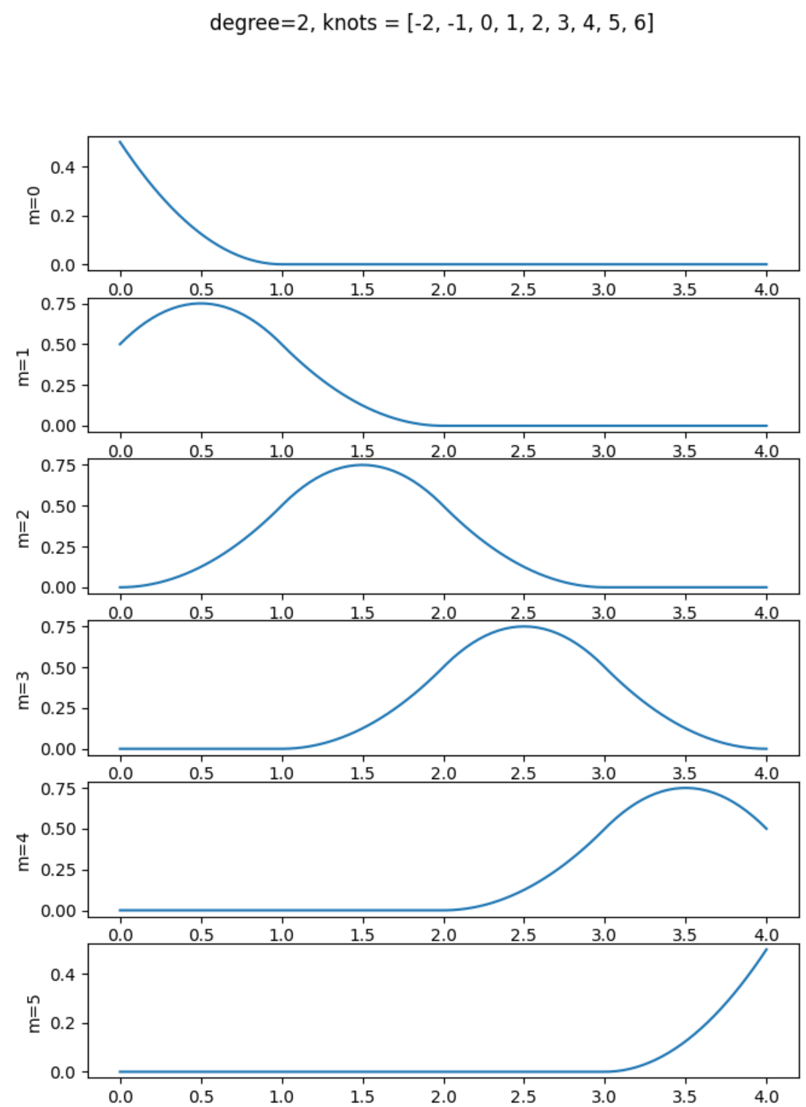

Figure 5: Second degree spline basis with extra knot

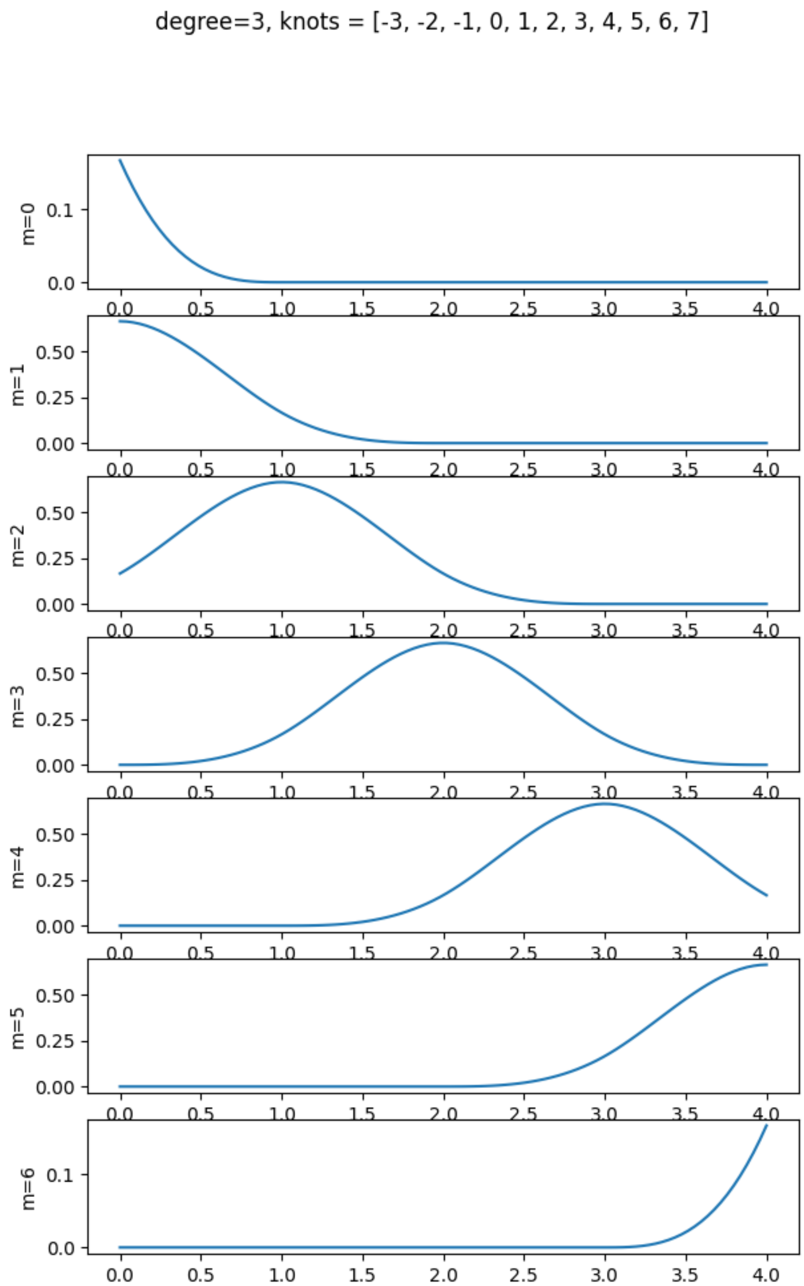

Figure 6: Third degree spline basis

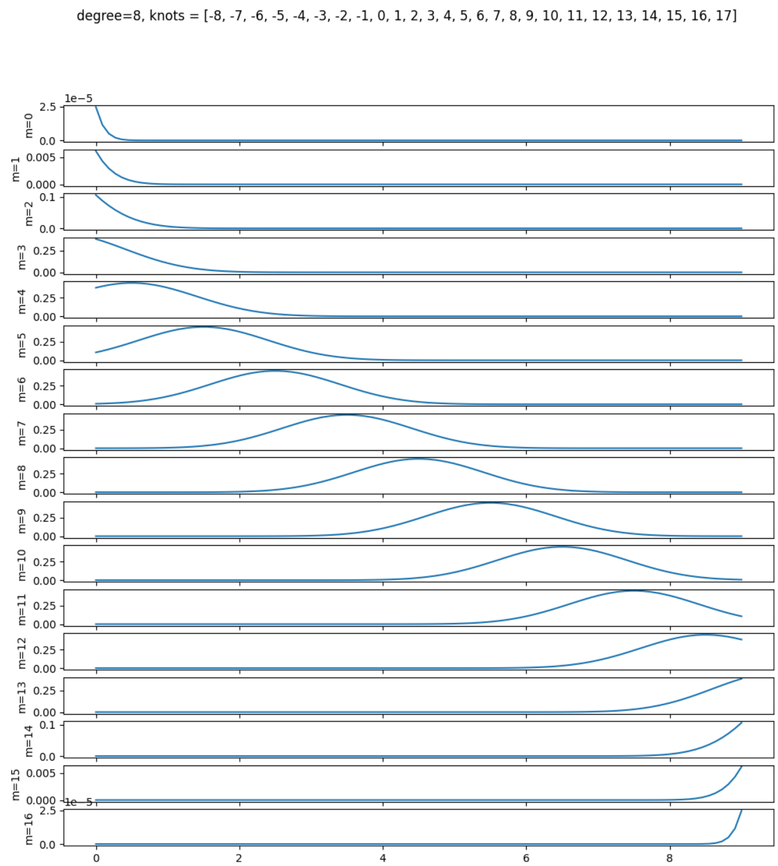

Figure 7: Eight degree spline basis

Shift and Scale Invariance of the Knot Sequence

Lemma 1.1. *Given an arbitrary knot sequence*

$$\mathbf{t} = [\tau_0, \tau_1, \ldots \tau_T],$$

an arbitrary time shift Δ, and the one-sequence $\mathbf{1} \mathrel{\hat{=}} [1, 1, \ldots, 1]$, the B-spline basis functions are shift invariant in the sense that if

$$\mathbf{t} + \Delta\mathbf{1} = [\tau_0 + \Delta, \tau_1 + \Delta, \ldots \tau_T + \Delta]$$

then

$$b_j^d(t, \mathbf{t} + \Delta\mathbf{1}) = b_j^d(t - \Delta, \mathbf{t}),$$

i.e., shifting the knot sequence forward in time is identical to shifting the original B-spline forward in time.

Proof. When the degree is $d = 0$, then equation (2) gives

$$\begin{aligned} b_j^0(t, \mathbf{t} + \Delta\mathbf{1}) &= \begin{cases} 1 & \text{if } \tau_j + \Delta \le t \le \tau_{j+1} + \Delta \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } \tau_j \le t - \Delta \le \tau_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ &= b_j^0(t - \Delta, \mathbf{t}). \end{aligned}$$

Assuming that the statement holds when the order is $d \ge 1$, we get from Equation (3) that

$$\begin{aligned} b_j^d(t, \mathbf{t} + \Delta\mathbf{1}) &= \frac{t - \tau_j - \Delta}{\tau_{j+d} + \Delta - \tau_j - \Delta} b_j^{d-1}(t, \mathbf{t} + \Delta\mathbf{1}) + \frac{\tau_{j+d+1} + \Delta - t}{\tau_{j+d+1} + \Delta - \tau_{j+1} - \Delta} b_{j+1}^{d-1}(t, \mathbf{t} + \Delta\mathbf{1}) \\ &= \frac{(t - \Delta) - \tau_j}{\tau_{j+d} - \tau_j} b_j^{d-1}(t - \Delta, \mathbf{t}) + \frac{\tau_{j+d+1} - (t - \Delta)}{\tau_{j+d+1} - \tau_{j+1}} b_{j+1}^{d-1}(t - \Delta, \mathbf{t}) \\ &= b_j^d(t - \Delta, \mathbf{t}). \end{aligned}$$

The proof therefore follows by induction. □

Lemma 1.2. *Given an arbitrary knot sequence*

$$\mathbf{t} = [\tau_0, \tau_1, \ldots \tau_T]$$

and an arbitrary scaling constant $\alpha \in \mathbb{R}$, the B-spline basis functions are scale invariant in the sense that if

$$\alpha\mathbf{t} = [\alpha\tau_0, \alpha\tau_1, \ldots \alpha\tau_T]$$

then

$$b_j^d(t, \alpha\mathbf{t}) = b_j^d(t/\alpha, \mathbf{t}),$$

i.e., scaling the knot sequence by α is identical to time scaling the original B-spline by $\frac{1}{\alpha}$.

Proof. When the degree is $d = 0$, then equation (2) gives

$$\begin{aligned} b_j^0(t, \alpha\mathbf{t}) &= \begin{cases} 1 & \text{if } \alpha\tau_j \leq t \leq \alpha\tau_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } \tau_j \leq \frac{t}{\alpha} \leq \tau_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ &= b_j^0(t/\alpha, \mathbf{t}). \end{aligned}$$

Assuming that the statement holds when the degree is $d \geq 1$, we get from Equation (3) that

$$\begin{aligned} b_j^d(t, \alpha\mathbf{t}) &= \frac{t - \alpha\tau_j}{\alpha\tau_{j+d} - \alpha\tau_j} b_j^{d-1}(t, \alpha\mathbf{t}) + \frac{\alpha\tau_{j+d+1} - t}{\alpha\tau_{j+d+1} - \alpha\tau_{j+1}} b_{j+1}^{d-1}(t, \alpha\mathbf{t}) \\ &= \frac{(t/\alpha) - \tau_j}{\tau_{j+d} - \tau_j} b_j^{d-1}(t/\alpha, \mathbf{t}) + \frac{\tau_{j+d+1} - (t/\alpha)}{\tau_{j+d+1} - \tau_{j+1}} b_{j+1}^{d-1}(t/\alpha, \mathbf{t}) \\ &= b_j^d(t/\alpha, \mathbf{t}). \end{aligned}$$

The proof therefore follows by induction. □

1.2 Natural, Uniform, and Clamped B-Splines

In the previous section we saw that there is a relationship between the knot point vector $\mathbf{t}$ and the basis functions. In this section we will define three different types of knot vectors that will be important in robotic applications: natural knot vectors, uniform knot vectors, and clamped knot vectors.

Definition 1.3. *We say that the knot vector* $\mathbf{t}_M^d = [\tau_0, \ldots, \tau_{M+1+2d}]$ *is a* natural *knot vector if* $M > d$ *and its first and last* d *knot values are non-decreasing* ($i < j \implies \tau_i \leq \tau_j$)*, and its middle* $M+1$ *terms are strictly increasing* ($i < j \implies \tau_i < \tau_j$)*, i.e., it has form*

$$\mathbf{t}_M^d = [\underbrace{t_{-d}, \ldots, t_{-1},}_{d-terms} \ \underbrace{t_0, t_1, \ldots, t_M,}_{(M+1)-terms} \ \underbrace{t_{M+1}, \ldots, t_{M+d}}_{d-terms}].$$

If the knot vector is natural, then we say that the B-spline (1) *is natural, and we restrict the times where the spline is valid to the interval* $t \in [t_0, t_M] = [\tau_d, \tau_{d+M}] = span(\mathbf{t})$.

For example, the following are natural knot vectors:

$$\mathbf{t}_{M=3}^{d=2} = [-2, -1.1, \vdots\ 0, 0.1, 2.5, 3, \vdots\ 4, 4], \quad \text{span}(\mathbf{t}_3^2) = [0, 3]$$

$$\mathbf{t}_{M=5}^{d=3} = [-6, -6, -4, \vdots\ -3, -2.1, -1, 0.5, 1, 2.5, \vdots\ 3, 3, 5], \quad \text{span}(\mathbf{t}_5^3) = [-3, 2.5]$$

$$\mathbf{t}_{M=5}^{d=3} = [-\frac{3}{5}, -\frac{3}{5}, -\frac{3}{5}, \vdots\ 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, \vdots\ \frac{6}{5}, \frac{7}{5}, \frac{8}{5}], \quad \text{span}(\mathbf{t}_5^3) = [0, 1].$$

Definition 1.4. *We say that the knot vector* $\mathbf{t}_M^d = [\tau_0, \ldots, \tau_{M+1+2d}]$ *is a* uniform *knot vector if it is a natural knot vector and the knot values are equally spaced, i.e., for* $\Delta > 0$ *it has form*

$$\mathbf{t}_M^d = [\underbrace{t_0 - d\Delta, \ldots, t_0 - \Delta,}_{d-terms} \underbrace{t_0, t_0 + \Delta, t_0 + 2\Delta, \ldots, t_0 + M\Delta,}_{(M+1)-terms} \underbrace{t_0 + (M+1)\Delta, \ldots, t_0 + (M+d)\Delta}_{d-terms}].$$

If the knot vector is uniform, then we say that the B-spline (1) *is uniform, and we restrict the times where the spline is valid to the interval* $t \in [t_0, t_M] = [\tau_d, \tau_{d+M}] = span(\mathbf{t}_M^d)$.

For example, the following are uniform knot vectors:

$$\mathbf{t}_{M=3}^{d=2} = [-2, -1, \vdots\, 0, 1, 2, 3, \vdots\, 4, 5], \quad \Delta = 1, \quad \text{span}(\mathbf{t}_3^2) = [0, 3],$$

$$\mathbf{t}_{M=5}^{d=3} = [-6, -5, -4, \vdots\, -3, -2, -1, 0, 1, 2, \vdots\, 3, 4, 5], \quad \Delta = 1, \quad \text{span}(\mathbf{t}_5^3) = [-3, 2],$$

$$\mathbf{t}_{M=5}^{d=3} = [-\frac{3}{5}, -\frac{2}{5}, -\frac{1}{5}, \vdots\, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, \vdots\, \frac{6}{5}, \frac{7}{5}, \frac{8}{5}], \quad \Delta = \frac{1}{5}, \quad \text{span}(\mathbf{t}_5^3) = [0, 1].$$

Definition 1.5. *We say that the knot vector* $\mathbf{t}_M^d$ *is* clamped *if it is a natural knot vector and the first and last d-terms are repeated, i.e. it has form*

$$\mathbf{t}_M^d = [\underbrace{t_0, t_0, \ldots, t_0,}_{d-terms} \underbrace{t_0, t_1, t_2, \ldots, t_M,}_{(M+1)-terms} \underbrace{t_M, \ldots, t_M}_{d-terms}],$$

where $t_i < t_{i+1}$ *for* $i = 0, \ldots, M-1$. *If* $t_0, \ldots, t_M$ *are uniformly spaced, then we say that the knot vector is* uniform and clamped. *If the knot vector is clamped, then we say that the B-spline* (1) *is clamped, and we restrict the times where the spline is valid to the interval* $t \in [t_0, t_M] = [\tau_d, \tau_{d+M}] = span(\mathbf{t}_M^d)$.

For example, in the following, the first and third knot vectors are uniform and clamped, while the second knot vector is clamped:

$$\mathbf{t}_{M=3}^{d=2} = [0, 0, \vdots\, 0, 1, 2, 3, \vdots\, 3, 3], \quad \Delta = 1, \quad \text{span}(\mathbf{t}_3^2) = [0, 3],$$

$$\mathbf{t}_{M=5}^{d=3} = [-3, -3, -3, \vdots\, -3, -2, -1.5, 0.1, 1.6, 2, \vdots\, 2, 2, 2], \quad \text{span}(\mathbf{t}_5^3) = [-3, 2],$$

$$\mathbf{t}_{M=5}^{d=3} = [0, 0, 0, \vdots\, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, \vdots\, 1, 1, 1], \quad \Delta = \frac{1}{5}, \quad \text{span}(\mathbf{t}_5^3) = [0, 1].$$

The length of natural knot vectors is

$$\text{length}(\mathbf{t}_M^d) = d + M + 1 + d = M + 2d + 1.$$

Given the shift and scale invariance properties shown in Lemmas 1.1 and 1.2, when the knot sequence is uniform, then without loss of generality we can use the knot vectors

$$\mathbf{t}_M^d = [-(d), \ldots, -1, 0, 1, 2, \ldots, M, M+1, \ldots, M+d]. \tag{5}$$

Similarly, when the knot sequence is uniform and clamped, then without loss of generality we can use

$$\mathbf{t}_M^d = [\underbrace{0, 0, \ldots, 0}_{d-\text{terms}}, 0, 1, 2, \ldots, M, \underbrace{M, \ldots, M}_{d-\text{terms}}]. \tag{6}$$

In both cases $\text{span}(\mathbf{t}_M^d) = [0, M]$.

In the case of uniform knot vectors, all basis function are just shifted versions of a single "central" basis function.

Lemma 1.6. *Let $\mathbf{t}_M^d$ be the uniform knot vector defined in Equation (5). Then the basis function $b_d^d(t, \mathbf{t}_M^d)$ plays a central role in the sense that*

$$b_{d+m}^d(t, \mathbf{t}_M^d) = b_d^d(t-m, \mathbf{t}_M^d), \qquad m = -(d+1), \ldots, M-1.$$

Proof. From Equation (2) we get that

$$\begin{aligned} b_m^0(t, \mathbf{t}_M^d) &= \begin{cases} 1 & \text{if } m \le t \le m+1 \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } 0 \le t-m \le 1 \\ 0 & \text{otherwise} \end{cases} \\ &= b_0^0(t-m, \mathbf{t}_M^d). \end{aligned}$$

Suppose that $b_{d-1+m}^{d-1}(t, \mathbf{t}_M^d) = b_{d-1}^{d-1}(t-m, \mathbf{t}_M^d)$, then

$$\begin{aligned} b_{d+m}^d(t, \mathbf{t}_M^d) &= \frac{t-(d+m)}{m+2d-(m+d)} b_{d+m}^{d-1}(t, \mathbf{t}_M^d) + \frac{m+2d+1-t}{m+2d+1-(m+d+1)} b_{d+m+1}^{d-1}(t, \mathbf{t}_M^d), \\ &= \frac{(t-m)-d)}{d} b_d^{d-1}(t-m; \mathbf{t}_M^d) + \frac{2d+1-(t-m)}{d} b_{d+1}^{d-1}(t-m; \mathbf{t}_M^d), \\ &= b_d^d(t-m; \mathbf{t}_M^d). \end{aligned}$$

Therefore, the lemma holds by induction. $\square$

1.3 Finite-Support Property

One of the most important properties of B-splines is the so-called finite-support property, which states that at any instant of time, only a few control points influence the B-spline. In this section, we formalize this property. We begin by showing that for any time $t \in \text{span}(\mathbf{t}_M^d)$, there are only $d+1$ non-zero basis functions.

Lemma 1.7. *Let $\mathbf{t}_M^d = [t_{-d}, \ldots, t_{-1}, t_0, \ldots, t_M, \ldots t_{M+d}]$ be a natural knot vector. Then for $j \leq d$, and $0 \leq m \leq M + 2(d-j)$, we have that*

$$b_m^j(t, \mathbf{t}_M^d) \neq 0 \quad \iff \quad t \in [t_{m-d}, t_{m+j-d}]. \tag{7}$$

Proof. Let

$$\begin{aligned} \mathbf{t}_M^d &= [\tau_0, \ldots, \tau_{d-1}, \tau_d, \ldots, \tau_{M+d}, \ldots, \tau_{M+2d}] \\ &= [t_{-d}, \ldots, t_{-1}, t_0, \ldots, t_M, \ldots t_{M+d}], \end{aligned}$$

where it is clear that $\tau_m = t_{m-d}$ for $m = 0, \ldots, M+2d$. Then from Equations (2) and (3) we have that $b_m^0(t, \mathbf{t}_M^d) \neq 0 \iff t \in [\tau_m, \tau_{m+1}] = [t_{m-d}, t_{m+1-d}]$. By induction, it is straightforward to show that Equation (7) holds since incrementing the degree by one, expands the non-zero support by one time interval. □

Corollary 1.8. *Let $\mathbf{t}_M^d = [t_{-d}, \ldots, t_{-1}, t_0, \ldots, t_M, \ldots t_{M+d}]$ be a natural knot vector. Then for any $t \in span(\mathbf{t}_M^d)$, there are exactly $j+1$ non-zero basis functions of degree $j \leq d$ at that time. In particular, if $t \in [t_s, t_{s+1}] \subset span(\mathbf{t}_M^d)$, then*

$$b_m^j(t, \mathbf{t}_M^d) \neq 0 \iff m \in [s, s+j].$$

Corollary 1.9. *Let $\mathbf{t}_M^d = [t_{-d}, \ldots, t_{-1}, t_0, \ldots, t_M, \ldots t_{M+d}]$ be a natural knot vector. If $0 \leq s \leq M-1$ is an integer, then for any $t \in [t_s, t_{s+1}] \subset span(\mathbf{t}_M^d)$,*

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d) = \sum_{m=s}^{s+d} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d).$$

In other words, over the interval $t \in [t_s, t_{s+1}]$ there are only $d+1$ control points that influence $\mathbf{p}(t)$, namely $\{\mathbf{c}_s, \ldots, \mathbf{c}_{s+d}\}$.

Proof. The proof follows directly from Lemma 1.10. □

In the special case of uniform and clamped knot vectors, we get the following corollary.

Corollary 1.10. *Let $\mathbf{t}_M^d$ be the uniform (possibly clamped) knot vector defined in Equation (6). Then there are exactly $M+d$ non-zero basis function of degree d, namely $b_m^d(t, \mathbf{t}_M^d)$, $m = 0, \ldots, M+d-1$. Furthermore, the basis of support for $b_m^d(t, \mathbf{t}_M^d)$, i.e., the time interval where $b_m^d(t, \mathbf{t}_M^d)$ is non-zero is given by*

$$b_m^d(t, \mathbf{t}_M^d) \neq 0 \quad \textit{if} \quad \begin{cases} t \in [0, m+1] & 0 \leq m \leq d \\ t \in [m-d, m+1] & d+1 \leq m \leq M-1 \\ t \in [m-d, M] & M \leq m \leq M+d-1 \end{cases}.$$

Proof. Follows directly from Lemma 1.10 and the definition of uniform knot vectors. □

A graphical depiction of the region of support for the set of basis functions $\{b_m^d(t, \mathbf{t}_M^d)\}_{m=0}^{M+d-1}$ is shown in Figure 8 for a uniform clamped knot vector with $d = 3$ and $M = 8$.

$$\mathbf{t}_8^3 = [0, 0, 0, \vdots 0, 1, 2, 3, 4, 5, 6, 7, 8, \vdots 8, 8, 8]$$

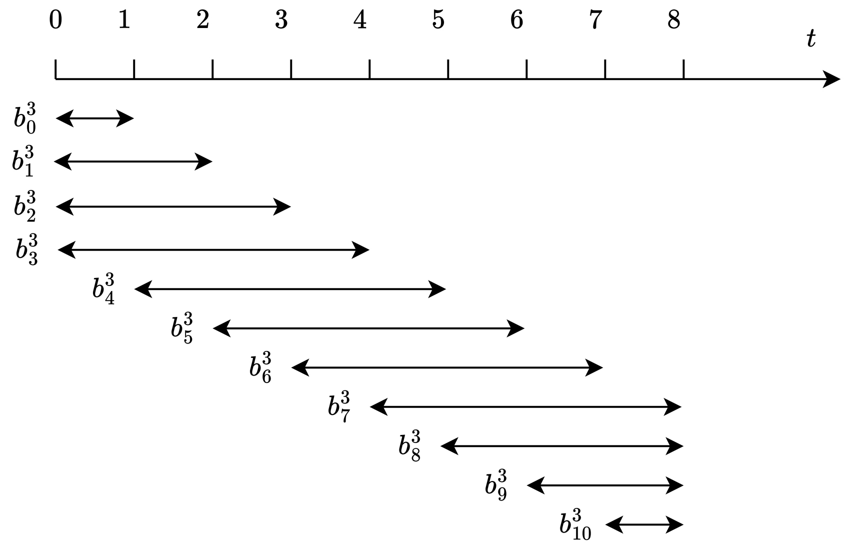

Figure 8: Region of support for the B-spline basis functions of degree $d = 3$ with uniform clamped knot vector $\mathbf{t}_8^d$.

1.4 Convex Hull Property

Another important property of B-splines, is that the spline $\mathbf{p}(t)$ is contained in the convex hull of its supporting control points. In this section we formalize the convex hull property.

Definition 1.11. *We say that the vector* $\mathbf{x}$ *is in the convex hull of the vectors* $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ *if*

$$\mathbf{x} = \sum_{j=1}^{n} \alpha_j \mathbf{q}_j, \qquad \textit{where} \qquad \sum_{j=1}^{n} \alpha_j = 1.$$

We first show that at any instant of time $t \in \text{span}(\mathbf{t}_M^b)$, the b-spline basis functions sum to unity.

Lemma 1.12. *Let* $\mathbf{t}_M^d$ *be a natural knot vector. Then for any instant of time* $t \in \textit{span}(\mathbf{t})$*, the basis vectors at that time sum to unity. In particular, if* $t \in [t_j, t_{j+1}] \subset \textit{span}(\mathbf{t})$*, for* $j \in [0, M-1]$*, then*

$$\sum_{m=0}^{M+d-1} b_m^d(t, \mathbf{t}_M^d) = \sum_{m=j}^{j+d} b_m^d(t, \mathbf{t}_M^d) = 1.$$

Proof. The proof follows from Lemma 1.10, a careful, but straight-forward accounting using Equations (2) and (3). $\square$

Lemma 1.13. *Let $\mathbf{t}_M^d$ be a natural knot vector. If $t \in [t_j, t_{j+1}] \subset span(\mathbf{t})$, for $j \in [0, M-1]$, then the B-spline*

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d) = \sum_{m=j}^{j+d} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d)$$

is in the convex hull of the control points $\{\mathbf{c}_j, \ldots, \mathbf{c}_{j+d}\}$.

Proof. The lemma follows from Lemma 1.12 and Corollary 1.9. □

Lemma 1.13 is an important result for path planning since we can guarantee collision-free paths by simply checking that the convex hull of control points is collision free.

SciPy BSpline library The SciPy library has a B-spline library.

The following commands will create a cubic spline.

```
import numpy as np
from scipy.interpolate import BSpline


def uniform_clamped_knots(k, M, t0=np.inf, tf=np.inf):
    # k is the order, M is the number of time intervals
    knots = [0] * k + list(range(0, M + 1)) + [M] * k
    knots = np.asarray(knots)  # convert knots to an NP array
    if t0 != np.inf:
        if (tf != np.inf) & (tf > t0):
            knots = (tf-t0)/M * knots
        knots = knots + t0
    return knots

t0 = 1 # initial time
tf = 5 # final time
k = 3  # spline order
M = 3  # number of time intervals
knots = uniform_clamped_knots(k=order, M=M, t0=t0, tf=tf)
# need M+k control points
ctrl_pts = np.array([[0, 0, 0],
                     [0, 1, 0],
                     [0, 0, 1],
                     [0, 1, 1],
                     [1, 1, 0],
                     [1, 1, 1]])
spl = BSpline(t=knots, c=ctrl_pts, k=order)
plot_spline(spl)
```

Where `plot_spline` is given below.

```
from math import ceil
from scipy.interpolate import BSpline
import matplotlib.pyplot as plt

def plot_spline(spl):
    t0 = spl.t[0]  # first knot is t0
    tf = spl.t[-1]  # last knot is tf
    # number of points in time vector so spacing is 0.01
    N = ceil((tf - t0)/0.01)
    t = np.linspace(t0, tf, N)  # time vector
    position = spl(t)
    # 3D trajectory plot
    fig = plt.figure(1)
    ax = fig.add_subplot(111, projection='3d')
    # plot control points (convert YX(-Z) -> NED)
    ax.plot(spl.c[:, 1], spl.c[:, 0], -spl.c[:, 2],
            '-o', label='control points')
    # plot spline (convert YX(-Z) -> NED)
    ax.plot(position[:, 1], position[:, 0], -position[:, 2],
            'b', label='spline')
    ax.legend()
    ax.set_xlabel('x', fontsize=16, rotation=0)
    ax.set_ylabel('y', fontsize=16, rotation=0)
    ax.set_zlabel('z', fontsize=16, rotation=0)
    #ax.set_xlim3d([-10, 10])
    plt.show()
```

The resulting spline is shown in Figure 9.

1.5 Derivatives of B-splines

To simplify notation we will stack the basis function in a vector as

$$\mathbf{b}_M^d(t) \mathrel{\hat{=}} \begin{pmatrix} b_0^d(t, \mathbf{t}_M^d) \\ b_1^d(t, \mathbf{t}_M^d) \\ \vdots \\ b_{M+d-1}^d(t, \mathbf{t}_M^d) \end{pmatrix}, \tag{8}$$

and the control points as a matrix

$$\mathbf{C} = \begin{pmatrix} \mathbf{c}_0 & \mathbf{c}_1 & \dots & \mathbf{c}_{M+d-1} \end{pmatrix}$$

allowing Equation (1) to be written as

$$\mathbf{p}(t) = \mathbf{C}\mathbf{b}_M^d(t), \qquad t\ in \text{span}(\mathbf{t}_M^d).$$

We begin this section by noting that the basis functions and the knot vector do not have to be of the same degree. In fact, if the knot vector has higher

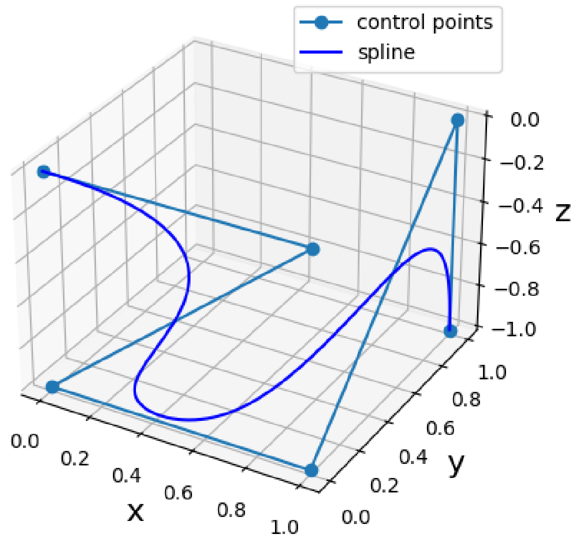

Figure 9: Example spline curve

degree than the basis functions, then the first several basis functions will simply be zero. For example, let $\mathbf{t}_3^2 = [0, 0, \vdots 0, 1, 2, 3, \vdots 3, 3]$, then from Equation (2) $b_0^0(t, \mathbf{t}_3^2)$, $b_1^0(t, \mathbf{t}_3^2)$, $b_5^0(t, \mathbf{t}_3^2)$, $b_6^0(t, \mathbf{t}_3^2)$, $b_0^1(t, \mathbf{t}_3^2)$, and $b_5^1(t, \mathbf{t}_3^2)$ are equal to zero since the knot intervals in the denominator for those functions are zero. The remaining basis functions are shown in Figure 10. If on the other hand, the knot vector has one degree lower, i.e., $\mathbf{t}_3^1 = [0, \vdots 0, 1, 2, 3, \vdots 3]$, then the basis vectors are also shown to the right in Figure 10. It is clear that reducing the degree of the knot vector by one, shifts the index of the basis vectors by one. We can formalize these observations in the following lemma.

Lemma 1.14. *For natural knot vectors* $\mathbf{t}_M^d \in \mathbb{R}^{M+2d+1}$ *and* $\mathbf{t}_M^{d+1} \in \mathbb{R}^{M+2d-1}$ *we have for* $t \in span(\mathbf{t}_M^d) = span(\mathbf{t}_M^{d+1})$,

$$b_m^j(t, \mathbf{t}_M^d) = b_{m+1}^j(t, \mathbf{t}_M^{d+1}), \qquad j \le d, \qquad m = 0, \ldots, M+1+2(d-j).$$

Using vector notation we have that

$$\underbrace{\mathbf{b}_M^j(t, \mathbf{t}_M^d)}_{(M+1+2d)\times 1} = S_{M+1+2d} \underbrace{\mathbf{b}_M^j(t, \mathbf{t}_M^{d+1})}_{(M+1+2(d+1))\times 1},$$

where

$$S_N \doteq \begin{bmatrix} \mathbf{0}_{N\times 1}, & \mathbf{I}_{N\times N}, & \mathbf{0}_{N\times 1} \end{bmatrix}.$$

Proof. Let

$$\begin{aligned}
\mathbf{t}_M^{d+1} &= [\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_{d+1}, \ldots, \hat{\tau}_{M+d+2}, \ldots, \hat{\tau}_{M+2d+2}, \hat{\tau}_{M+2d+3}] \\
&= [t_{-d-1}, t_{-d}, \ldots, t_0, \ldots, t_M, \ldots, t_{M+d}, t_{M+d+1}] \\
\mathbf{t}_M^d &= [\tau_0, \ldots, \tau_d, \ldots, \tau_{M+d+1}, \ldots, \tau_{M+2d+1}] \\
f &= [t_{-d}, \ldots, t_0, \ldots, t_M, \ldots, t_{M+d}],
\end{aligned}$$

where it is clear that $\tau_j = \hat{\tau}_{j+1}$. Therefore, from Equations (2)we have

$$\begin{aligned}
b_m^0(t, \mathbf{t}_M^d) &= \begin{cases} 1 & \text{if } \tau_m \le t \le \tau_{m+1} \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 & \text{if } \hat{\tau}_{m+1} \le t \le \hat{\tau}_{m+2} \\ 0 & \text{otherwise} \end{cases} \\
&= b_{m+1}^0(t, \mathbf{t}_M^{d+1}).
\end{aligned}$$

Equations (3) and (4) follow by similar manipulation. □

We have the following lemma which gives the general formula for the derivative of the spline basis.

Lemma 1.15. *If $\mathbf{t}_M^d$ is a natural knot vector, then the ℓ^{th} derivative of the degree d spline function with $0 \le \ell \le d+1$, is given by*

$$\frac{d^\ell}{dt^\ell} b_m^d(t, \mathbf{t}_M^d) = d \left(\frac{\frac{d^{\ell-1}}{dt^{\ell-1}} b_m^{d-1}(t, \mathbf{t}_M^d)}{\tau_{m+d} - \tau_m} - \frac{\frac{d^{\ell-1}}{dt^{\ell-1}} b_{m+1}^{d-1}(t, \mathbf{t}_M^d)}{\tau_{m+d+1} - \tau_{m+1}} \right).$$

The proof of Lemma 1.15 is in [**?**].

Using vector notation, we have the following formula for the derivative of a spline function.

Lemma 1.16. *Given the natural spline function of degree d*

$$\mathbf{p}(t) = \mathbf{C}\mathbf{b}_M^d(t), \quad t \in \mathit{span}(\mathbf{t}_M^d)$$

we have that

$$\frac{d\mathbf{p}}{dt}(t) = \mathbf{C} D_M^d \mathbf{b}_M^{d-1}(t), \quad t \in \mathit{span}(\mathbf{t}_M^{d-1})$$

where D_M^d is the $(M+d) \times (M+d-1)$ derivative matrix given by

$$D_M^d = - \begin{bmatrix} \bar{D}_M^d \\ \mathbf{0}_{1\times(M+d-1)} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{1\times(M+d-1)} \\ \bar{D}_M^d \end{bmatrix}, \tag{9}$$

where

$$\bar{D}_M^d = \mathit{diag}\left(\frac{d}{t_{-d+1} - t_1}, \frac{d}{t_{-d+2} - t_2}, \ldots, \frac{d}{t_{M+2d-2} - t_{M+d-2}} \right), \tag{10}$$

is an $(M+d-1) \times (M+d-2)$ diagonal matrix.

Proof. Let

$$\begin{aligned}
\mathbf{t}_M^d &= [\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_d, \ldots, \hat{\tau}_{M+d}, \ldots, \hat{\tau}_{M+2d-1}, \hat{\tau}_{M+2d}] \\
&= [t_{-d}, t_{-d+1} \ldots, t_0, \ldots, t_M, \ldots, t_{M+d-1}, t_{M+d}] \\
\mathbf{t}_M^{d-1} &= [\tau_0, \ldots, \tau_{d-1}, \ldots, \tau_{M+d-1}, \ldots, \tau_{M+2d-2}] \\
&= [t_{-d+1}, \ldots, t_0, \ldots, t_M, \ldots, t_{M+d-1}],
\end{aligned}$$

where it is clear that $\tau_j = \hat{\tau}_{j+1}$, for $j = 0, \ldots, M + 2d - 2$.

Noting that

$$\mathbf{p}(t) = \mathbf{C}\mathbf{b}_M^d(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d),$$

we have that

$$\frac{d\mathbf{p}}{dt} = \sum_{m=0}^{M+d-1} \mathbf{c}_m \frac{db_m^d(t, \mathbf{t}_M^d)}{dt}.$$

From Lemma 1.15 we get

$$\begin{aligned}
\frac{d\mathbf{p}}{dt} &= \sum_{m=0}^{M+d-1} \mathbf{c}_m(d) \left(\frac{b_m^{d-1}(t, \mathbf{t}_M^d)}{\hat{\tau}_{m+d} - \hat{\tau}_m} - \frac{b_{m+1}^{d-1}(t, \mathbf{t}_M^d)}{\hat{\tau}_{m+d+1} - \hat{\tau}_{m+1}} \right) \\
&= \sum_{m=0}^{M+d-1} \mathbf{c}_m \left[\left(\frac{d}{\hat{\tau}_{m+d} - \hat{\tau}_m} \right) b_m^{d-1}(t, \mathbf{t}_M^d) - \left(\frac{d}{\hat{\tau}_{m+d+1} - \hat{\tau}_{m+1}} \right) b_{m+1}^{d-1}(t, \mathbf{t}_M^d) \right] \\
&= \left(\frac{d}{\hat{\tau}_d - \hat{\tau}_0} \right) \mathbf{c}_0 b_0^{d-1}(t, \mathbf{t}_M^d) + \sum_{m=1}^{M+d-1} \left(\frac{d}{\hat{\tau}_{m+d} - \hat{\tau}_m} \right) (\mathbf{c}_m - \mathbf{c}_{m-1})\, b_m^{d-1}(t, \mathbf{t}_M^d) \\
&\quad - \left(\frac{d}{\hat{\tau}_{M+2d} - \hat{\tau}_{M+d}} \right) \mathbf{c}_{M+d-1} b_{M+d}^{d-1}(t, \mathbf{t}_M^d).
\end{aligned}$$

From Lemma 1.7 we have that $b_0^{d-1}(t, \mathbf{b}_M^d)$ and $b_{M+d}^{d-1}(t, \mathbf{b}_M^d)$ are zero on $\mathrm{span}(\mathbf{t}_M^d)$, which implies that

$$\begin{aligned}
\frac{d\mathbf{p}}{dt} &= \sum_{m=1}^{M+d-1} \left(\frac{d}{\hat{\tau}_{m+d} - \hat{\tau}_m} \right) (\mathbf{c}_m - \mathbf{c}_{m-1})\, b_m^{d-1}(t, \mathbf{t}_M^d) \qquad (11) \\
&= \sum_{m=0}^{M+d-2} \left(\frac{d}{\hat{\tau}_{m+1+d} - \hat{\tau}_{m+1}} \right) (\mathbf{c}_{m+1} - \mathbf{c}_m)\, b_{m+1}^{d-1}(t, \mathbf{t}_M^d) \\
&= \sum_{m=0}^{M+d-2} \left(\frac{d}{\tau_{m+d} - \tau_m} \right) (\mathbf{c}_{m+1} - \mathbf{c}_m)\, b_m^{d-1}(t, \mathbf{t}_M^{d-1}) \\
&= \mathbf{C} D_M^d \mathbf{b}_M^{d-1}(t).
\end{aligned}$$

□

Note that if $\mathbf{t}_M^d$ is the uniform knot vector in Equation (5), then

$$\bar{D}_M^d = I_{M+d-2},$$

since the interval in the denominator is always equal to d. Similarly, if $\mathbf{t}_M^d$ is the uniform clamped knot vector in Equation (6), then

$$\bar{D}_M^d = \text{diag}\left(\frac{d}{1}, \frac{d}{2}, \dots, \frac{d}{d-1}, \underbrace{1, \dots, 1}_{M-d}, \frac{d}{d-1}, \dots, \frac{d}{2}, \frac{d}{1}\right).$$

From Lemma 1.16 we can derive a number of useful results, which we summarize in the corollary below.

Corollary 1.17. *Given the natural/uniform/clamped B-spline function of degree d,*

$$\mathbf{p}(t) = \mathbf{C}\mathbf{b}_M^d(t), \quad t \in \mathit{span}(\mathbf{t}_M^d)$$

we can make the following statements:

(i) *The derivative $\frac{d\mathbf{p}}{dt}$ is a natural/uniform/clamped clamped B-spline function of degree $d-1$.*

(ii) *The control points of $\frac{d\mathbf{p}}{dt}$ are given by*

$$C^{'} \mathrel{\hat{=}} CD_M^d = \begin{pmatrix} \frac{d}{t_{-d+1}-t_1}(\mathbf{c}_1 - \mathbf{c}_0)^\top \\ \vdots \\ \frac{d}{t_{M+2d-2}-t_{M_d-2}}(\mathbf{c}_{M+d-1} - \mathbf{c}_{M+d-2})^\top \end{pmatrix}^\top \mathrel{\hat{=}} \begin{pmatrix} \mathbf{c}_0^{'\top} \\ \vdots \\ \mathbf{c}_{M+d-2}^{'\top} \end{pmatrix}^\top.$$

(iii) *The ℓ^{th} derivative of $\mathbf{p}(t)$ for $0 \leq \ell \leq d-1$ is given by*

$$\begin{aligned} \frac{d^\ell \mathbf{p}}{dt^\ell} &= CD_M^d D_M^{d-1} \dots D_M^{d-\ell} \mathbf{b}_M^{d-\ell}(t), \quad t \in \mathit{span}(\mathbf{t}_M^d) \\ &= C^{(\ell)} \mathbf{b}_M^{d-\ell}(t), \quad t \in \mathit{span}(\mathbf{t}_M^d) \\ &= C\boldsymbol{\psi}^{(\ell)}(t), \quad t \in \mathit{span}(\mathbf{t}_M^d) \end{aligned}$$

where the control points of the $(d-\ell)^{th}$ degree spline are given by

$$C^{(\ell)} = CD_M^d D_M^{d-1} \dots D_M^{d-\ell-1}$$

or alternatively, the ℓ^{th} derivative of the basis vector $\mathbf{b}_M^d(t)$ is given by

$$\boldsymbol{\psi}^{(\ell)} \mathrel{\hat{=}} \frac{d^\ell \mathbf{b}_M^d(t)}{dt^\ell} = D_M^d D_M^{d-1} \dots D_M^{d-\ell} \mathbf{b}_M^{d-\ell}(t), \quad t \in \mathit{span}(\mathbf{t}_M^d).$$

For uniform clamped B-splines, we have the following.

Corollary 1.18. *Given the uniform clamped B-spline function of degree d*

$$\mathbf{p}(t) = \mathbf{C}\mathbf{b}_M^d(t), \quad t \in [0, M]$$

we can make the following statements:

(i) *The control points of $\frac{d\mathbf{p}}{dt}$ are given by*

$$C^{'} \mathrel{\hat{=}} CD_M^k = \begin{pmatrix} \frac{d}{1}(\mathbf{c}_1 - \mathbf{c}_0)^\top \\ \vdots \\ \frac{d}{d-1}(\mathbf{c}_{d-2} - \mathbf{c}_{d-3})^\top \\ (\mathbf{c}_{d-1} - \mathbf{c}_{d-2})^\top \\ \vdots \\ (\mathbf{c}_{M+1} - \mathbf{c}_M)^\top \\ \frac{d}{d-1}(\mathbf{c}_{M+2} - \mathbf{c}_{M+1})^\top \\ \vdots \\ \frac{d}{1}(\mathbf{c}_{M+d-1} - \mathbf{c}_{M+d-2})^\top \end{pmatrix}^\top \mathrel{\hat{=}} \begin{pmatrix} \mathbf{c}_0^{'\top} \\ \vdots \\ \mathbf{c}_{M+d-2}^{'\top} \end{pmatrix}^\top.$$

(ii) *The derivative of $\mathbf{p}(t)$ at $t = 0$ is given by the difference of the first two control points as*

$$\frac{d\mathbf{p}}{dt}(0) = (d)(\mathbf{c}_1 - \mathbf{c}_0).$$

(iii) *The derivative of $\mathbf{p}(t)$ at $t = M$ is given by the difference of the last two control points as*

$$\frac{d\mathbf{p}}{dt}(M) = (d)(\mathbf{c}_{M+d-1} - \mathbf{c}_{M+d-2}).$$

(iv) *If the desired B-spline trajectory with $d \geq 3$ has the following desired end-point conditions:*

$$\begin{aligned} \textit{Initial position:} \quad & \mathbf{p}(0) \overset{des}{=} \mathbf{p}_0 \\ \textit{Final position:} \quad & \mathbf{p}(M) \overset{des}{=} \mathbf{p}_f \\ \textit{Initial velocity:} \quad & \frac{d\mathbf{p}}{dt}(0) \overset{des}{=} \mathbf{v}_0 \\ \textit{Final velocity:} \quad & \frac{d\mathbf{p}}{dt}(M) \overset{des}{=} \mathbf{v}_f, \\ \textit{Initial acceleration:} \quad & \frac{d^2\mathbf{p}}{dt^2}(0) \overset{des}{=} \mathbf{a}_0 \\ \textit{Final acceleration:} \quad & \frac{d^2\mathbf{p}}{dt^2}(M) \overset{des}{=} \mathbf{a}_f, \end{aligned}$$

then the first and last three control points satisfy

$$
\begin{aligned}
\mathbf{c}_0 &= \mathbf{p}_0 \\
\mathbf{c}_1 &= \mathbf{p}_0 + \frac{1}{d}\mathbf{v}_0 \\
\mathbf{c}_2 &= \mathbf{p}_0 + \frac{3}{d}\mathbf{v}_0 + \frac{2}{(d)(d-1)}\mathbf{a}_0 \\
\mathbf{c}_{M+d-3} &= \mathbf{p}_f - \frac{3}{d}\mathbf{v}_f + \frac{2}{(d)(d-1)}\mathbf{a}_f \\
\mathbf{c}_{M+d-2} &= \mathbf{p}_f - \frac{1}{d}\mathbf{v}_f \\
\mathbf{c}_{M+d-1} &= \mathbf{p}_f.
\end{aligned}
$$

2 Path Constraints

2.1 Constant Velocity Trajectories

Suppose that the B-spline trajectory of degree d

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{t}_M^d)$$

has been planned so that the trajectory is collision free, but that in creating the collision-free path the control points have been moved around. The velocity along the trajectory will vary with time based on the control points $\mathbf{c}_m$. The question that we will address in this section is whether the knot points can be adjusted so that the trajectory is constant velocity. In this case, we assume that $\mathbf{c}_m$, $m = 0, \ldots, M + d$ are fixed.

Lemma 2.1. *Given the control points $\mathbf{c}_m$, $m = 0, \ldots, M_d$, the initial time t_0, and the desired speed v^{des}*

$${}^{d}_{M} = [\tau_0, \ldots, \tau_d, \ldots \tau_{M+d}, \ldots, \tau_{M+2d}] = [\underbrace{t_0, \ldots, t_0}_{d+1 terms}, \ldots, \underbrace{t_M, \ldots, t_M}_{d+1 terms}],$$

where

$$\tau_m = \tau_{m-d} + dv^{des} \left\| \mathbf{c}_{m+1} - \mathbf{c}_m \right\|, \qquad m = 1, \ldots, M + d - 2$$

then $\left\| \mathbf{p} \right\|(t) = v^{des}$ for all $t \in span({}^{d}_{M})$.

Proof. From Equation (11) we have that

$$\frac{d\mathbf{p}}{dt} = \sum_{m=1}^{M+d-1} \left(\frac{d}{\tau_{m+d} - \tau_m} \right) (\mathbf{c}_m - \mathbf{c}_{m-1})\, b_m^{d-1}(t, \mathbf{t}_M^d).$$

$\square$

2.2 Curvature constraints

If $N > 4$, then $\Phi^\top$ has a non-trivial null space that can be exploited to satisfy additional constraints. As an example, we may want to minimize the curvature of the path.

The curvature of $p(\sigma)$ is defined by the formula

$$\kappa(\sigma) = \frac{p^{'}(\sigma) \times p^{''}(\sigma)}{\|p^{'}(\sigma)\|^3}.$$

RWB: Figure out how this constraints the control points.

3 Minimum Snap Trajectories

Because multirotor dynamics are differentially flat, their motion can be completely parametrized by trajectories in position and heading. It is argued in (Mellinger & Kumar, 2011) [**?**] that since acceleration is dependent on third derivative of position, and torque is dependent on the derivative of heading, that smooth trajectories for quadrotors should minimize the fourth derivative (snap) of the position spline, and the second derivative of the yaw spline. In this section, will show to derive quadratic cost functions on the spline coefficients that minimize the appropriate derivative.

Let the position and heading trajectories be defined by B-splines of degree d over the time interval $t \in [0, M]$ as

$$\hat{\mathbf{p}}(t) = \mathbf{C}_p b_M^d(t)$$
$$\hat{\psi}(t) = \mathbf{C}_\psi b_M^d(t),$$

where $\mathbf{C}_p \in \mathbb{R}^{3\times M+d}$, and $\mathbf{C}_\psi \in \mathbb{R}^{1\times M+d}$. As shown in Lemma 1.18, the fourth derivative of position and second derivative of heading are given by

$$\hat{\mathbf{p}}^{(4)}(t) = \mathbf{C}_p D_M^d D_M^{d-1} D_M^{d-2} D_M^{d-3} \mathbf{b}_M^{d-4}(t)$$
$$\hat{\psi}^{(2)}(t) = \mathbf{C}_\psi D_M^k D_M^{k-1} \mathbf{b}_M^{k-2}(t).$$

Defining

$$S_M^{d,j} \doteq D_M^d D_M^{d-1} \cdots D_M^{d-j}$$

gives

$$\hat{\mathbf{p}}^{(4)}(t) = \mathbf{C}_p S_M^{d,3} \mathbf{b}_M^{d-4}(t)$$
$$\hat{\psi}^{(2)}(t) = \mathbf{C}_\psi S_M^{d,1} \mathbf{b}_M^{d-2}(t).$$

The normed integral square of the fourth derivative of $\mathbf{p}(t)$ is given by

$$\begin{aligned}
\int_0^M \left\|\mathbf{p}^{(4)}(t)\right\|^2 dt &= \int_0^M \mathbf{p}^{(4)\top}(t)\mathbf{p}^{(4)}(t)\, dt \\
&= \int_0^M \mathbf{b}_M^{d-4}(t)^\top {S_M^{d,3}}^\top \mathbf{C}_p^\top \mathbf{C}_p S_M^{d,3} \mathbf{b}_M^{d-4}(t)\, dt \\
&= \operatorname{trace}\left(\int_0^M \mathbf{b}_M^{d-4}(t)^\top {S_M^{d,3}}^\top \mathbf{C}_p^\top \mathbf{C}_p S_M^{d,3} \mathbf{b}_M^{d-4}(t)\, dt\right) \\
&= \operatorname{trace}\left(\int_0^M \mathbf{C}_p S_M^{d,3} \mathbf{b}_M^{d-4}(t) \mathbf{b}_M^{d-4}(t)^\top {S_M^{d,3}}^\top \mathbf{C}_p^\top\, dt\right) \\
&= \operatorname{trace}\left(\mathbf{C}_p S_M^{d,3} \int_0^M \mathbf{b}_M^{d-4}(t) \mathbf{b}_M^{d-4}(t)^\top dt \;\; {S_M^{d,3}}^\top \mathbf{C}_p^\top\right).
\end{aligned}$$

Define

$$W_M^{d,j} \doteq S_M^{d,j} \int_0^M \mathbf{b}_M^{d-j}(t) \mathbf{b}_M^{d-j}(t)^\top dt \;\; {S_M^{d,j}}^\top,$$

and note that $W_M^{d,j}$ are constant matrices that can be pre-computed and stored in memory, then

$$\begin{aligned}
\int_0^M \left\|\mathbf{p}^{(4)}(t)\right\|^2 dt &= \operatorname{trace}\left(\mathbf{C}_p W_M^{d,4} \mathbf{C}_p^\top\right) \\
\int_0^M \left|\psi^{(2)}(t)\right|^2 dt &= \operatorname{trace}\left(\mathbf{C}_\psi W_M^{d,1} \mathbf{C}_\psi^\top\right).
\end{aligned}$$

The initial and final conditions place constraints on the control points as shown in Corollary 1.18. These conditions can be stated as matrix equality constraints on the control points. For example, when the initial and final position, velocity, and acceleration are specified, we have from Corollary 1.18 that

$$\begin{aligned}
\mathbf{c}_0 &= \mathbf{p}_0 \\
\mathbf{c}_1 &= \mathbf{p}_0 + \frac{1}{d}\mathbf{v}_0 \\
\mathbf{c}_2 &= \mathbf{p}_0 + \frac{3}{d}\mathbf{v}_0 + \frac{2}{(d)(d-1)}\mathbf{a}_0 \\
\mathbf{c}_{M+d-3} &= \mathbf{p}_f - \frac{3}{d}\mathbf{v}_f + \frac{2}{(d)(d-1)}\mathbf{a}_f \\
\mathbf{c}_{M+d-2} &= \mathbf{p}_f - \frac{1}{d}\mathbf{v}_f \\
\mathbf{c}_{M+d-1} &= \mathbf{p}_f.
\end{aligned}$$

which can be written in matrix form as

$$
\begin{aligned}
&\underbrace{\begin{pmatrix}\mathbf{c}_0 & \dots \mathbf{c}_{M+d-1}\end{pmatrix}}_{\mathbf{C}_p}\underbrace{\begin{pmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_{M+d-2} & \mathbf{e}_{M+d-1} & \mathbf{e}_{M+d}\end{pmatrix}}_{U_1} \\
&\qquad = \underbrace{\begin{pmatrix}\mathbf{p}_0 & \mathbf{v}_0 & \mathbf{a}_0 & \mathbf{a}_f & \mathbf{v}_f & \mathbf{p}_f\end{pmatrix}}_{A_p}\underbrace{\begin{pmatrix}1 & 1 & 1 & 0 & 0 & 0\\ 0 & \frac{1}{d} & \frac{3}{d} & 0 & 0 & 0\\ 0 & 0 & \frac{2}{(d)(d-1)} & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{2}{(d)(d-1)} & 0 & 0\\ 0 & 0 & 0 & \frac{-3}{d} & \frac{-1}{d} & 0\\ 0 & 0 & 0 & 1 & 1 & 1\end{pmatrix}}_{B^{d,3}}.
\end{aligned}
$$

The minimum-snap problem without obstacles, can therefore be written as

$$
\begin{aligned}
&\min_{\mathbf{C}_p} \operatorname{trace}\left(\mathbf{C}_p W_M^{d,4}\mathbf{C}_p^\top\right) \\
&\text{subject to } \ \mathbf{C}_p U_1 = A_p B^{d,3}.
\end{aligned}
\tag{12}
$$

3.1 Minimum snap trajectories without obstacles

Without obstacles, the minimum-snap problem has an analytic solution as derived in the next theorem.

Theorem 3.1. *The optimization problem given in Equation* (12) *has an analytic solution given by*

$$
\mathbf{C}_p^\star = A_p B^{d,3} U_1^\top \left(I - W_M^{d,4} U_2 (U_2^\top W_M^{d,4} U_2)^{-1} U_2^\top\right), \tag{13}
$$

where

$$
U_2 = \left(\mathbf{e}_4, \dots, \mathbf{e}_{M+d-3}\right).
$$

Proof. We begin by noting that $U_1^\top U_1 = I_{6\times 6}$ and that $U_2^\top U_1 = 0$. Let $\check{\mathbf{C}}_p = A_p B^{d,3} U_1^\top + Z U_2^\top$ and note that

$$
\begin{aligned}
\check{\mathbf{C}}_p U_1 &= \left(A_p B^{d,3} U_1^\top + Z U_2^\top\right) U_1 \\
&= A_p B^{d,3} U_1^\top U_1 + Z U_2^\top U_1 \\
&= A_p B^{d,3}.
\end{aligned}
$$

Therefore $\check{\mathbf{C}}_p$ satisfies the inequality constraints and implies that the optimal solution has the form of

$$
\mathbf{C}_p^* = A_p B^{d,3} U_1^\top + Z^* U_2^\top.
$$

We have therefore transformed the constrained optimization problem in (12) to the unconstrained optimization problem

$$\begin{aligned}
&\min_Z \operatorname{trace}\left(\left(A_p B^{d,3} U_1^\top + Z U_2^\top\right) W_M^{d,4} \left(A_p B^{d,3} U_1^\top + Z U_2^\top\right)^\top\right) \\
&= \min_Z \operatorname{trace}\Big(A_p B^{d,3} U_1^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top + Z U_2^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top \\
&\qquad\qquad + A_p B^{d,3} U_1 \top W_M^{d,4} U_2 Z^\top + Z U_2^\top W_M^{d,4} U_2 Z^\top\Big) \\
&= \min_Z \operatorname{trace}\left(A_p B^{d,3} U_1^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top + 2 Z U_2^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top + Z U_2^\top W_M^{d,4} U_2 Z^\top\right),
\end{aligned}$$

where we have used the fact that $\operatorname{trace}\left(M^\top\right) = \operatorname{trace}(M)$. Letting

$$J = \operatorname{trace}\left(A_p B^{d,3} U_1^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top + 2 Z U_2^\top W_M^{d,4} U_1 B^{d,4\top} A_p^\top + Z U_2^\top W_M^{d,4} U_2 Z^\top\right)$$

and recalling that for matrix equations

$$\begin{aligned}
\frac{\partial}{\partial X} \operatorname{trace}(XM) &= M^\top \\
\frac{\partial}{\partial X} \operatorname{trace}\left(X M X^\top\right) &= X(M + M^\top),
\end{aligned}$$

we get that

$$\frac{\partial J}{\partial Z} = 2 A_p B^{d,3} U_1^\top W_M^{d,4} U_2 + 2 Z U_2^\top W_M^{d,4} U_2,$$

where we have used the fact that $W_M^{d,j}$ is symmetric. Setting $\frac{\partial J}{\partial Z}$ to zero and solving for the optimal Z gives

$$Z^* = -A_p B^{d,3} U_1^\top W_M^{d,4} U_2 (U_2^\top W_M^{d,4} U_2)^{-1}.$$

Therefore

$$\begin{aligned}
\mathbf{C}_p^* &= A_p B^{d,3} U_1^\top + Z^* U_2^\top \\
&= A_p B^{d,3} U_1^\top + \left(-A_p B^{d,3} U_1^\top W_M^{d,4} U_2 (U_2^\top W_M^{d,4} U_2)^{-1}\right) U_2^\top \\
&= A_p B^{d,3} U_1^\top \left(I - W_M^{d,4} U_2 (U_2^\top W_M^{d,4} U_2)^{-1} U_2^\top\right).
\end{aligned}$$

□

This is a very nice result in the sense that the matrix

$$Q_M^{d,4} = B^{d,3} U_1^\top \left(I - W_M^{d,4} U_2 (U_2^\top W_M^{d,4} U_2)^{-1} U_2^\top\right)$$

is a constant, problem independent matrix that can be pre-computed and stored in memory, and the optimal control points can be computed simply from the initial and final conditions as

$$\mathbf{C}_p^* = A_p Q_M^{d,4}.$$

For the heading spline, we assume constraints on the initial and final heading as

$$\psi_0 = \hat{\psi}(0) = \mathbf{C}_\psi b^d_M(0) = c_{\psi,0}$$
$$\psi_f = \hat{\psi}(M) = \mathbf{C}_\psi b^d_M(M) = c_{\psi,M+d-1},$$

which can be written in matrix form as

$$\underbrace{\begin{pmatrix} c_{\psi,0} & \dots c_{\psi,M+d-1} \end{pmatrix}}_{\mathbf{C}_\psi} \underbrace{\begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_{M+d} \end{pmatrix}}_{U_{1,\psi}} = \underbrace{\begin{pmatrix} \psi_0 & \psi_f \end{pmatrix}}_{A_\psi} \underbrace{I_{2\times 2}}_{B^{d,2}}.$$

Using the same logic as above, we have the following theorem for the heading spline.

Theorem 3.2. *The solution to the optimization problem*

$$\begin{aligned} &\min_{\mathbf{C}_\psi} \mathit{trace}\left(\mathbf{C}_\psi W^{d,2}_M \mathbf{C}_\psi^\top\right) \\ &\mathit{subject\ to}\ \ \mathbf{C}_\psi U_{1,\psi} = A_\psi \end{aligned} \tag{14}$$

is given by

$$\mathbf{C}^\star_\psi = A_\psi Q^{d,2}_M, \tag{15}$$

where

$$Q^{d,2}_M = U_{1,\psi}^\top \left(I - W^{d,2}_M U_{2,\psi}(U_{2,\psi}^\top W^{d,2}_M U_{2,\psi})^{-1} U_{2,\psi}^\top\right)$$
$$U_{2,\psi} = \left(\mathbf{e}_2, \dots, \mathbf{e}_{M+d-1}\right).$$

3.2 Minimum snap trajectories with obstacles

4 Path Planning using a Digital Elevation Map

Add a discussion similar to this paper.

citeManyamCasbeerWeintraub21: This paper uses a delaney trangulation to parameterize the free space, and then moves control points within the delayney triangles to satisfy constraints.

[**?**]: This paper gets a quick and dirty path using the fast marching method, and then creates boxes around the desired trajectory produces a safe path corridor. The b-spline paths are then optimized to stay within the corridor.

[**?**]: This paper shows how to get quick and dirty paths using the Fast Marching Method (FMM).

[**?**]: Collision avoidance constraints, as describe in [**?**]. If the obstacles are ellipsoids: Suppose that the world is modeled by a set of ellipsoidal obstacles

$$\mathcal{O}_i = \{x \in \mathbf{R}^3 : x^\top P_i x \leq 1\}$$

and the total set of obstacles in the world is

$$\mathcal{W} = \bigcup_{i=1}^{M} \mathcal{O}_i.$$

In this case, we want to ensure that none of the control points is in an obstacle, i.e., that

This paper shows how to formulate the obstacle avoidance problem as a convex problem using dual variables.

5 B-Splines and Solutions of Ordinary Differential Equations

Given the (single input single output) state space system

$$\dot{x} = Ax + Bu \tag{16}$$

$$y = Cx \tag{17}$$

with initial condition $x(t_0) = x_0$, the solution to the differential equation at time $t \geq t_0$ is given by

$$y(t) = Ce^{A(t-t_0)}x_0 + \int_{t_0}^{t} Ce^{A(t-\tau)}Bu(\tau)\,d\tau.$$

Suppose that the input is given by the spline function

$$u(t) = \sum_{j=0}^{n-1} c_j \phi_j(t) \mathrel{\hat{=}} \mathbf{c}^\top \boldsymbol{\phi}(t),$$

then the solution to the differential equation is given by

$$\begin{aligned} y(t) &= Ce^{A(t-t_0)}x_0 + \int_{t_0}^{t} Ce^{A(t-\tau)}B\left(\sum_{j=0}^{n-1} c_j\phi_j(\tau)\right)d\tau \\ &= Ce^{A(t-t_0)}x_0 + \sum_{j=0}^{n-1} c_j\left(\int_{t_0}^{t} Ce^{A(t-\tau)}B\phi_j(\tau)\,d\tau\right). \end{aligned}$$

Defining the function

$$\psi_j(t) \mathrel{\hat{=}} \int_{t_0}^{t} Ce^{A(t-\tau)}B\phi_j(\tau)\,d\tau$$

and the vector $\boldsymbol{\psi}(t) = (\psi_0(t), \ldots, \psi_{n-1}(t))^\top$, then

$$y(t) = Ce^{A(t-t_0)}x_0 + \mathbf{c}^\top\boldsymbol{\psi}(t). \tag{18}$$

Using this framework, we can solve a variety of open-loop control problem.

Example 5.1. *Given the dynamic system represented by Equations* (16)–(17)*, find $u(t)$ to transfer the system from $y_0 = Cx_0$, to a desired final output $y(t_f) = y_f$. From Equation* (18) *we have*

$$y_f = Ce^{A(t_f - t_0)}x_0 + \boldsymbol{\psi}(t_f)^\top \mathbf{c}.$$

If we assume that $\mathbf{c} = \zeta\boldsymbol{\psi}(t_f)$, where $\zeta \in \mathbb{R}$, then ζ satisfies

$$\begin{aligned} y_f - Ce^{A(t_f - t_0)}x_0 &= \boldsymbol{\psi}(t_f)^\top \boldsymbol{\psi}(t_f)\zeta \\ \implies \zeta &= \frac{y_f - Ce^{A(t_f - t_0)}x_0}{\|\boldsymbol{\psi}(t_f)\|^2}. \end{aligned}$$

Therefore, the spline coefficients that transfer the system between the two desired outputs are

$$\mathbf{c} = \left(\frac{\boldsymbol{\psi}(t_f)}{\|\boldsymbol{\psi}(t_f)\|^2}\right)\left(y_f - Ce^{A(t_f - t_0)}x_0\right).$$

Example 5.2. *Assume that the initial condition is x_0, and assume that the control objective is to follow a reference signal $y_r(t)$. Find the control signal that minimizes the integral of the squared error:*

$$\int_0^{t_f} \|y(\tau) - y_r(\tau)\|^2 \, d\tau. \tag{19}$$

In this case we have

$$\begin{aligned} \int_0^{t_f} \|y(\tau) - y_r(\tau)\|^2 \, d\tau &= \int_0^{t_f} (y(\tau) - y_r(\tau))^\top (y(\tau) - y_r(\tau)) d\tau \\ &= \int_0^{t_f} (\boldsymbol{\psi}(\tau)^\top \mathbf{c} - y_r(\tau))^\top (\boldsymbol{\psi}(\tau)^\top \mathbf{c} - y_r(\tau)) d\tau \\ &= \int_0^{t_f} \left(\mathbf{c}^\top \boldsymbol{\psi}(\tau)\boldsymbol{\psi}(\tau)^\top \mathbf{c} - 2y_r(\tau)\boldsymbol{\psi}(\tau)^\top \mathbf{c} + \|y_r(\tau)\|^2\right) d\tau \\ &= \mathbf{c}^\top \left(\int_0^{t_f} \boldsymbol{\psi}(\tau)\boldsymbol{\psi}(\tau)^\top d\tau\right)\mathbf{c} - 2\left(\int_0^{t_f} y_r(\tau)\boldsymbol{\psi}(\tau)^\top d\tau\right)\mathbf{c} \\ &\quad + \left(\int_0^{t_f} \|y_r(\tau)\|^2 \, d\tau\right) \\ &= \mathbf{c}^\top A\mathbf{c} - 2b^\top \mathbf{c} + d \end{aligned}$$

where

$$\begin{aligned} A &= \int_0^{t_f} \boldsymbol{\psi}(\tau)\boldsymbol{\psi}(\tau)^\top d\tau \\ b &= \int_0^{t_f} y_r(\tau)\boldsymbol{\psi}(\tau) d\tau \\ d &= \int_0^{t_f} \|y_r(\tau)\|^2 \, d\tau. \end{aligned}$$

The (unconstrained) control input that minimizes Equation (19) *satisfies*

$$2A\mathbf{c} - 2b = 0.$$

Therefore, the spline coefficients are given by

$$\mathbf{c} = A^{-1}b = \left(\int_0^{t_f} \boldsymbol{\psi}(\tau)\boldsymbol{\psi}(\tau)^\top d\tau\right)^{-1} \left(\int_0^{t_f} y_r(\tau)\boldsymbol{\psi}(\tau) d\tau\right).$$

5

6 B-Spline Planning for Chains of Integrators

7 B-Splines on Lie Groups

8 B-Splines Surfaces

Suppose that the objective is to create a one dimensional surface over a two dimensional domain. For example, a terrain map is a one dimensional surface of altitudes over the north-east plane. In this section we will explore the use of B-splines to accomplish this objective. Many of the ideas and notation in this section come from [**?**].

If $\mathbf{t}_M^k$ and $\mathbf{t}_N^k$ are two different k^{th} order uniform clamped knot vectors of length M and N understood to define knots in the x and y directions, then following Equation (**??**), a clamped uniform B-spline surface is defined as

$$S(x,y) \triangleq \sum_{m=0}^{M+k-1} \sum_{n=0}^{N+k-1} c_{m,n} b_m^k(x; \mathbf{t}_M^k) b_n^k(y; \mathbf{t}_N^k), \quad x \in [0,M], y \in [0,N], \tag{20}$$

where $c_{m,n}$, $m = 0, \dots, M+k-1$, $n = 0, \dots, N+k-1$ are the control points of the surface. Defining the matrix $\mathbf{C} = \{c_{m,n}\}$, and defining the spatial variable $\mathbf{s} = (s_1, s_2)^\top$, we can write Equation (20) as

$$S(\mathbf{s}) = \mathbf{b}_M^k(s_1)^\top \mathbf{C} \mathbf{b}_N^k(s_2), \quad \mathbf{s} \in [0,M] \times [0,N], \tag{21}$$

This equation is linear in the spline coefficients $\mathbf{C}$. In other words, if $S_1(\mathbf{s}) = \mathbf{b}_M^k(s_1)^\top \mathbf{C}_1 \mathbf{b}_N^k(s_2)$, and $S_1(\mathbf{s}) = \mathbf{b}_M^k(s_1)^\top \mathbf{C}_2 \mathbf{b}_N^k(s_2)$ are two different spline surfaces, then

$$S(\mathbf{s}) \triangleq \alpha S_1(\mathbf{s}) + \beta S_2(\mathbf{s}) = \mathbf{b}_M^k(s_1)^\top \left(\alpha \mathbf{C}_1 + \beta \mathbf{C}_2\right) \mathbf{b}_N^k(s_2),$$

is also a spline surface.

Suppose that instead of defining the surface over the set $[0,M] \times [0,N]$ we instead would like to define the surface over the set $[\underline{X}, \overline{X}] \times [\underline{Y}, \overline{Y}]$. Given the shift and scaling properties is Lemmas 1.1 and 1.2 we have

$$S(\mathbf{s}) = \mathbf{b}_M^k \left(\frac{M(s_1 - \underline{X})}{\overline{X} - \underline{X}}\right)^\top \mathbf{C} \mathbf{b}_N^k \left(\frac{N(s_2 - \underline{Y})}{\overline{Y} - \underline{Y}}\right), \quad \mathbf{s} \in [\underline{X}, \overline{X}] \times [\underline{Y}, \overline{Y}]. \tag{22}$$

The Python code below plots a randomly defined terrain map over the domain $[-\pi, \pi] \times [-2\pi, 2\pi]$.

```
import numpy as np
import splipy as sp
import matplotlib.pyplot as plt
from bspline_tools import uniform_clamped_knots

def draw_random_surface(order=2, M=10, N=10,
                        Xmin=-3.14159, Xmax=3.14159,
                        Ymin=-2*3.14159, Ymax=2*3.14159):
    # define the spline surface
    knots_x = uniform_clamped_knots(k=order, M=M)
    knots_y = uniform_clamped_knots(k=order, M=N)
    basis_x = sp.BSplineBasis(order + 1, knots_x)
    basis_y = sp.BSplineBasis(order + 1, knots_y)
    C = np.random.rand(M + order, N + order)  # random control points
    surface = sp.Surface(basis_x, basis_y,
                         np.reshape(C, ((M + order) * (N + order), 1)))
    # plot the spline surface
    x = np.linspace(Xmin, Xmax, 10 * M)
    y = np.linspace(Ymin, Ymax, 10 * N)
    S = surface(M*(x-Xmin)/(Xmax-Xmin),
                N*(y-Ymin)/(Ymax-Ymin))  # surface points
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    plt.xlabel('x')
    plt.ylabel('y')
    x_pts, y_pts = np.meshgrid(x, y)  # grid points
    ax.plot_surface(x_pts, y_pts, S[:, :, 0])
    plt.show()
```

The result of running this code is shown in Figure 12

9 B-Splines and Occupancy Maps

Since B-spline surfaces are continuous the spatial variable $\mathbf{s} \in \mathbf{R}^2$, the occupancy map will also be continuous (as opposed to an occupancy grid). Following [**?**], at every point $\mathbf{s}$ we can define the discrete random variable $m(\mathbf{s})$ to be the occupancy of the map, where $m(\mathbf{s}) = 1$ indicates that a target is located at $\mathbf{s}$, and $m(\mathbf{s}) = 0$ indicates that a target is not located at $\mathbf{s}$. Similarly, define the discrete random variable $z(\mathbf{s})$ to be the measurement state, where $z(\mathbf{s}) = 1$ indicates that a target is detected at $\mathbf{s}$, and $z(\mathbf{s}) = 0$ indicates that a target is not detected at $\mathbf{s}$. Let $\mathbf{x}_t$ be the state of the UAV at time t, where the UAV state evolution equations are given by

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \eta_t$$

where $\eta_t \sim \mathcal{N}(0, Q)$. Let $fov(\mathbf{x}_t)$ be the area on the terrain what is seen by the camera when the state is $\mathbf{x}_t$, then the measurement model is given by

$$p(z(\mathbf{s}) = 1 \mid m(\mathbf{s}) = 1, \mathbf{x}) = \begin{cases} p_d & \mathbf{s} \in fov(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

$$p(z(\mathbf{s}) = 0 \mid m(\mathbf{s}) = 1, \mathbf{x}) = \begin{cases} 1 - p_d & \mathbf{s} \in fov(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases} \tag{24}$$

$$p(z(\mathbf{s}) = 1 \mid m(\mathbf{s}) = 0, \mathbf{x}) = \begin{cases} p_f & \mathbf{s} \in fov(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases} \tag{25}$$

$$p(z(\mathbf{s}) = 0 \mid m(\mathbf{s}) = 0, \mathbf{x}) = \begin{cases} 1 - p_f & \mathbf{s} \in fov(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}, \tag{26}$$

where p_d is the probability of detection, and p_f is the probability of false alarm.

Let $Z_t(\mathbf{s}) = \{z_t(\mathbf{s}), z_{t-1}(\mathbf{s}), \ldots, z_0(\mathbf{s})\}$ be the set of measurements of map location $\mathbf{s}$ over the time horizon $[0, t]$. Using Bayes's law we have

$$p(m(\mathbf{s}) = 1 \mid Z_t(\mathbf{s})) = \frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, Z_{t-1}(\mathbf{s}))p(m(\mathbf{s}) = 1 \mid Z_{t-1}(\mathbf{s}))}{p(z_t(\mathbf{s}) \mid Z_{t-1}(\mathbf{s}))}$$

$$p(m(\mathbf{s}) = 0 \mid Z_t(\mathbf{s})) = \frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, Z_{t-1}(\mathbf{s}))p(m(\mathbf{s}) = 0 \mid Z_{t-1}(\mathbf{s}))}{p(z_t(\mathbf{s}) \mid Z_{t-1}(\mathbf{s}))}$$

Defining the odds of a discrete random variable as

$$odds\,(x) \mathrel{\hat{=}} \frac{p(x = 1)}{p(x = 0)},$$

we get that

$$\begin{aligned} odds\,(m(\mathbf{s}) \mid Z_t(\mathbf{s}), \mathbf{x}_t) &= \frac{p(m(\mathbf{s}) = 1 \mid Z_t(\mathbf{s}), \mathbf{x}_t)}{p(m(\mathbf{s}) = 0 \mid Z_t(\mathbf{s}), \mathbf{x}_t)} \\ &= \frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, Z_{t-1}(\mathbf{s}), \mathbf{x}_t)\, p(m(\mathbf{s}) = 1) \mid Z_{t-1}(\mathbf{s}), \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, Z_{t-1}(\mathbf{s}), \mathbf{x}_t)\, p(m(\mathbf{s}) = 0) \mid Z_{t-1}(\mathbf{s}), \mathbf{x}_t)} \\ &= odds\,(m(\mathbf{s}) \mid Z_{t-1}(\mathbf{s}), \mathbf{x}_t)\, \frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, Z_{t-1}(\mathbf{s}), \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, Z_{t-1}(\mathbf{s}), \mathbf{x}_t)}. \end{aligned}$$

Assuming that the current measurement is conditionally independent of previous measurements we get

$$odds\,(m(\mathbf{s}) \mid Z_t(\mathbf{s}), \mathbf{x}_t) = odds\,(m(\mathbf{s}) \mid Z_{t-1}(\mathbf{s}), \mathbf{x}_t)\, \frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}.$$

Taking the natural logarithm of each side gives

$$\log odds\,(m(\mathbf{s}) \mid Z_t(\mathbf{s}), \mathbf{x}_t) = \log odds\,(m(\mathbf{s}) \mid Z_{t-1}(\mathbf{s}), \mathbf{x}_t) + \log\left(\frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}\right).$$

Defining the expected value (over random variable $\mathbf{x}_t$) as

$$S_t(\mathbf{s}) \mathrel{\mathring{=}} E\{\log odds\,(m(\mathbf{s}) \mid Z_t(\mathbf{s}), \mathbf{x}_t)\},$$

then the expected log-odds surface at time t is given by

$$S_t(\mathbf{s}) = S_{t-1}(\mathbf{s}) + \int \log \left[\frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}\right] p(\mathbf{x}_t) d\mathbf{x}_t.$$

Defining the measurement function as

$$U(z_t(\mathbf{s}), \mathbf{s}) \mathrel{\mathring{=}} \int \log \left[\frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}\right] p(\mathbf{x}_t) d\mathbf{x}_t, \tag{27}$$

then the log-odds update expression is

$$S_t(\mathbf{s}) = S_{t-1}(\mathbf{s}) + U(z_t(\mathbf{s}), \mathbf{s}). \tag{28}$$

Following [**?**] we will represent $S_t(\mathbf{s})$ using a B-spline surface as

$$S_t(\mathbf{s}) = \mathbf{b}_M^k(s_1)^\top C_t \mathbf{b}_N^k(s_2), \quad \mathbf{s} \in [0, M] \times [0, N]. \tag{29}$$

The probability that the target is located at $\mathbf{s}$ can be recovered from $S_t(\mathbf{s})$ using the formula

$$p(m(\mathbf{s}) = 1 \mid Z_t(\mathbf{s})) = \frac{e^{S_t(\mathbf{s})}}{1 + e^{S_t(\mathbf{s})}}.$$

Lemma 9.1. *Given the log-odds spline surface* (29)*, a measurement function* $U(z_t(\mathbf{s}), \mathbf{s})$*, and the log-odds update equation* (28)*, the optimal update for the spline coefficients after a measurement at* $\mathbf{s} \in [0, M] \times [0, N]$ *is given by*

$$\mathbf{C}_t = \mathbf{C}_{t-1} + \mathbf{B}_{M,N}^k(\mathbf{s}) U(z_t(\mathbf{s}), \mathbf{s}), \tag{30}$$

where

$$\mathbf{B}_{M,N}^k(\mathbf{s}) \mathrel{\mathring{=}} \frac{\mathbf{b}_M^k(s_1)\mathbf{b}_N^k(s_2)^\top}{\left\|\mathbf{b}_M^k(s_1)\right\|^2 \left\|\mathbf{b}_N^k(s_2)\right\|^2} \tag{31}$$

Proof. The proof follows the discussion given in [**?**, Section III.A]. Define the $vec\,(\cdot)$ operation on a matrix to be the operation of stacking all of the columns of a matrix into a single vector, i.e., $vec\,((\mathbf{c}_1, \ldots, \mathbf{c}_N)) = (\mathbf{c}_1^\top, \ldots, \mathbf{c}_N^\top)^\top$, and let $\otimes$ denote the Kronecker product. As shown in [**?**], if A, X, and B are compatible matrices, then $vec\,(AXB) = (B^\top \otimes A) vec\,(X)$. Accordingly, we can write the spline surface (29) as

$$\begin{aligned}
S_t(\mathbf{s}) &= \mathbf{b}_M^k(s_1)^\top \mathbf{C}_t \mathbf{b}_N^k(s_2) \\
&= vec\left(\mathbf{b}_M^k(s_1)^\top \mathbf{C}_t \mathbf{b}_N^k(s_2)\right) \\
&= \left[\mathbf{b}_N^k(s_2)^\top \otimes \mathbf{b}_M^k(s_1)^\top\right] vec\,(\mathbf{C}_t) \\
&= \boldsymbol{\phi}(\mathbf{s})^\top \mathbf{c}_t,
\end{aligned}$$

where $\boldsymbol{\phi}(\mathbf{s}) \doteq \mathbf{b}_N^k(s_2) \otimes \mathbf{b}_M^k(s_1)$, and $\mathbf{c}_t \doteq vec(\mathbf{C}_t)$.

Consider the log-odds update Equation (28), define the error equation at $\mathbf{s}$ to be

$$e(\mathbf{c}_t) \doteq \boldsymbol{\phi}(\mathbf{s})^\top \mathbf{c}_t - \boldsymbol{\phi}(\mathbf{s})^\top \mathbf{c}_{t-1} - U(z_t(\mathbf{s}), \mathbf{s}), \tag{32}$$

and consider the cost function $J(\mathbf{c}_t) = \frac{1}{2} e(\mathbf{c}_t)^2$. Using a gradient descent update law for $\mathbf{c}_t$ would result in

$$\begin{aligned}
\mathbf{c}_t &= \mathbf{c}_{t-1} - \mu \frac{\partial J}{\partial \mathbf{c}_t} \\
&= \mathbf{c}_{t-1} - \mu \frac{\partial e}{\partial \mathbf{c}_t} e(\mathbf{c}_t) \\
&= \mathbf{c}_{t-1} - \mu \boldsymbol{\phi}(\mathbf{s}) e(\mathbf{c}_t) \\
&= \mathbf{c}_{t-1} - \mu \boldsymbol{\phi}(\mathbf{s}) \left(\boldsymbol{\phi}(\mathbf{s})^\top \mathbf{c}_t - \boldsymbol{\phi}(\mathbf{s})^\top \mathbf{c}_{t-1} - U(z_t(\mathbf{s}), \mathbf{s}) \right).
\end{aligned}$$

Solving for $\mathbf{c}_t$ gives

$$\mathbf{c}_t = \mathbf{c}_{t-1} + \mu \left[I + \mu \boldsymbol{\phi}(\mathbf{s}) \boldsymbol{\phi}(\mathbf{s})^\top \right]^{-1} \boldsymbol{\phi}(\mathbf{s}) U(z_t(\mathbf{s}), \mathbf{s}), \tag{33}$$

where we note that $\left[I + \mu \boldsymbol{\phi}(\mathbf{s}) \boldsymbol{\phi}(\mathbf{s})^\top \right]$ is invertible for all $\mu > 0$ since $\boldsymbol{\phi}\boldsymbol{\phi}^\top$ is positive semi-definite. From the matrix inversion lemma we have that

$$\left[I + \mu \boldsymbol{\phi}(\mathbf{s}) \boldsymbol{\phi}(\mathbf{s})^\top \right]^{-1} = I - \mu \frac{\boldsymbol{\phi}(\mathbf{s}) \boldsymbol{\phi}(\mathbf{s})^\top}{1 + \mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2}.$$

Therefore Equation (33) becomes

$$\begin{aligned}
\mathbf{c}_t &= \mathbf{c}_{t-1} + \mu \left[I - \mu \frac{\boldsymbol{\phi}(\mathbf{s}) \boldsymbol{\phi}(\mathbf{s})^\top}{1 + \mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2} \right] \boldsymbol{\phi}(\mathbf{s}) U(z_t(\mathbf{s}), \mathbf{s}) \\
&= \mathbf{c}_{t-1} + \mu \boldsymbol{\phi}(\mathbf{s}) \left[1 - \mu \frac{\left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2}{1 + \mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2} \right] U(z_t(\mathbf{s}), \mathbf{s}) \\
&= \mathbf{c}_{t-1} + \left[\frac{\mu}{1 + \mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2} \right] \boldsymbol{\phi}(\mathbf{s}) U(z_t(\mathbf{s}), \mathbf{s}).
\end{aligned} \tag{34}$$

The error equation (32) then becomes

$$e(\mathbf{c}_t) = \left[\frac{\mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2}{1 + \mu \left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2} \right] U(z_t(\mathbf{s}), \mathbf{s}) - U(z_t(\mathbf{s}), \mathbf{s}).$$

Therefore, the squared error is minimized in the limit as $\mu \to \infty$. As $\mu \to \infty$, the spline coefficient update equation (34) becomes

$$\mathbf{c}_t = \mathbf{c}_{t-1} + \frac{\boldsymbol{\phi}(\mathbf{s})}{\left\| \boldsymbol{\phi}(\mathbf{s}) \right\|^2} U(z_t(\mathbf{s}), \mathbf{s}),$$

or in other words

$$vec\left(\mathbf{C}_t\right) = vec\left(\mathbf{C}_{t-1}\right) + \frac{\mathbf{b}_N^k(s_2) \otimes \mathbf{b}_M^k(s_1)}{\left\|\mathbf{b}_N^k(s_2) \otimes \mathbf{b}_M^k(s_1)\right\|^2} U(z_t(\mathbf{s}), \mathbf{s}).$$

Unstacking the vectors into matrices, and noting that

$$\mathbf{b}_N^k(s_2) \otimes \mathbf{b}_M^k(s_1) = vec\left(\mathbf{b}_M^k(s_1)\mathbf{b}_N^k(s_2)^\top\right)$$

and that

$$\left\|\mathbf{b}_N^k(s_2) \otimes \mathbf{b}_M^k(s_1)\right\|^2 = \left\|\mathbf{b}_M^k(s_1)\right\|^2 \left\|\mathbf{b}_N^k(s_2)\right\|^2,$$

results in Equation (30). □

Lemma 9.2. *Let $\mathbf{B}_{M,N}^k(\mathbf{s})$ be the $(M+k)\times(N+k)$ measurement update matrix defined in Equation (31). If $\mathbf{s} \in [m, m+1] \times [n, n+1] \subset [0, M] \times [0, N]$, then the only nonzero elements of $\mathbf{B}_{M,N}^k(\mathbf{s})$ is the $(k+1) \times (k+1)$ sub-matrix with upper-left index equal to $[m, n]$.*

Proof. The proof follows directly from Lemma 1.9. □

Lemma's 9.1 and 9.2 taken together imply that every measurement $U(z_t(\mathbf{s}), \mathbf{s})$ only requires the update of $(k+1)^2$ spline coefficients, and is therefore, extremely computationally efficient. For example, if the order of the splines is $k = 3$, then the measurement update is over a 4×4 submatrix, independent of the size of M and N, which for large worlds could be in tens of thousands. This property has obvious implications for cooperative control where vehicles can build cooperative maps with limited communication.

We now turn our attention to the computation of the update equation $U(z_t(\mathbf{s}), \mathbf{s})$ given by Equation (27) as

$$U(z_t(\mathbf{s}), \mathbf{s}) = \int_{\boldsymbol{\xi}\in\mathbb{R}^n} \log\left[\frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}\right] p_{X_t}(\boldsymbol{\xi}) d\boldsymbol{\xi},$$

where

$$p_{X_t}(\boldsymbol{\xi}) \doteq \frac{1}{\sqrt{2\pi^n |P_t|}} exp\left(\frac{1}{2}(\boldsymbol{\xi} - \mathbf{x}_t)^\top \mathbf{P}_t^{-1}(\boldsymbol{\xi} - \mathbf{x}_t)\right)$$

is the posterior probability density associated with state estimate of the aircraft, where $(\mathbf{x}_t, \mathbf{P}_t)$ are the mean and covariance given by, for example, an extended Kalman filter. From Equations (23)–(26) we get that

$$\log\left[\frac{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 1, \mathbf{x}_t)}{p(z_t(\mathbf{s}) \mid m(\mathbf{s}) = 0, \mathbf{x}_t)}\right] = \begin{cases} \log\left(\frac{p_d}{p_f}\right), & z_t(\mathbf{s}) = 1 \text{ and } \mathbf{s} \in fov(\mathbf{x}_t) \\ \log\left(\frac{1-p_d}{1-p_f}\right), & z_t(\mathbf{s}) = 0 \text{ and } \mathbf{s} \in fov(\mathbf{x}_t) \\ ? & \mathbf{s} \notin fov(\mathbf{x}_t) \end{cases} \cdot$$

Defining the set $fov^{-1}(\mathbf{s}) \doteq \{\boldsymbol{\xi} \in \mathbb{R}^n : \mathbf{s} \in fov(\boldsymbol{\xi})\}$ and defining

$$\bar{U}(\mathbf{s}) \doteq \int_{\boldsymbol{\xi}\in fov^{-1}(\mathbf{s})} p_{X_t}(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{35}$$

we get that the measurement update equation is given by

$$U(z_t(\mathbf{s}), \mathbf{s}) = \begin{cases} \bar{U}(\mathbf{s}) \log\left(\frac{p_d}{p_f}\right), & z_t(\mathbf{s}) = 1 \\ \bar{U}(\mathbf{s}) \log\left(\frac{1-p_d}{1-p_f}\right), & z_t(\mathbf{s}) = 0 \end{cases}.$$

The computation of $\bar{U}(\mathbf{s})$ will require a few assumptions, as well as numerical approximation of the integral. In approximating Equation (35) we will assume that the field of view of the camera is rectangular with field-of-view angles η_x, and η_y in the x and y axes of the camera frame. Figure 13 shows the physical geometry. We will assume that the camera is gimbaled to point toward the center of the earth, and that the ground is approximately flat. In that case, the physical footprint of the camera field of view will also be rectangular with physical dimensions given by $M_x = 2h\tan(\eta_x/2)$ and $M_y = 2h\tan(\eta_y/2)$, where $h = -p_d$ is the altitude of the aircraft. For the purposes of this section, we will assume that there is uncertainty in the north-east position of the aircraft, but that the altitude and attitude are known. If the aircraft is equipped with an altimeter and digital compass, then this is a reasonable assumption. Given the estimated state of the UAV $\mathbf{x}_t$ we can determine footprint of field-of-view on the ground. We assume that the target detection software can detect targets to some resolution in the camera, and divide the field of view into rectangular bins accordingly. Define N_x and N_y so that there are $2N_x + 1$ bins in the x-direction, and $2N_y + 1$ bins in the y-direction, as shown in Figure 14. We define the number of bins in this way to ensure an odd number in each direction. The physical size of each bin is given by

$$A = \left(\frac{M_x}{2N_x + 1}\right)\left(\frac{M_y}{2N_y + 1}\right) = \left(\frac{2h\tan(\eta_x/2)}{2N_x + 1}\right)\left(\frac{2h\tan(\eta_y/2)}{2N_y + 1}\right).$$

Suppose that the state of the UAV at time t is given by

$$\mathbf{x}_t = (p_n, p_e, p_d, v_n, v_e, v_d, \phi, \theta, \psi)^\top$$

then the position of the center of the $(i,j)^{th}$ bin is given by

$$\mathbf{s}_{i,j} = \begin{pmatrix} p_n \\ p_e \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \frac{M_x(h)}{2N_x+1} i \\ \frac{M_y(h)}{2N_y+1} j \end{pmatrix},$$

for $i = -N_x, \ldots, N_x$ and $j = -N_y, \ldots, N_y$.

Given a camera measurement, we will need to update the log-odds spline map for each $\mathbf{s}_{i,j}$ in the camera field of view. To compute $\bar{U}(\mathbf{s}_{i,j})$ requires the inverse field-of-view of $\mathbf{s}_{i,j}$. Under the assumption that the h and ψ are known, the inverse field-of-view will a rectangle on the altitude plane of the aircraft centered at $\mathbf{s}_{i,j}$ as shown in Figure 13. Using a similar binning notation for the inverse field-of-view, the north-east coordinates of the center of the $(m,n)^{th}$ bin of the inverse field of view is given by

$$
\begin{aligned}
&\mathbf{s}_{i,j} + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \frac{M_x(h)}{2N_x+1} m \\ \frac{M_y(h)}{2N_y+1} n \end{pmatrix} \\
=& \begin{pmatrix} p_n \\ p_e \end{pmatrix} + + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \frac{M_x(h)}{2N_x+1} (i+m) \\ \frac{M_y(h)}{2N_y+1} (j+n) \end{pmatrix} \\
=& \mathbf{s}_{i+m,j+n}
\end{aligned}
$$

for $(i,m) = -N_x, \ldots, N_x$ and $(j,n) = -N_y, \ldots, N_y$.

Defining the projection matrix $\mathbf{\Pi} = (\mathbf{I}_2, \mathbf{0})$, the projection of $\mathbf{x}_t$ onto the north-east plane is given by $\mathbf{\Pi x}_t$, and the associated error covariance, restricted to the error in the north-east coordinates is given by $\mathbf{\Pi P}_t \mathbf{\Pi}^\top$

The probability distribution over the projection of $\mathbf{x}$ onto the north-east plane is therefore given by

$$
p_{\mathbf{\Pi} X_t}(\boldsymbol{\sigma}) = \frac{1}{\sqrt{(2\pi)^2 |\mathbf{\Pi} P_t \mathbf{\Pi}^\top|}} \exp\left(\frac{1}{2} (\boldsymbol{\sigma} - \mathbf{\Pi x}_t)^\top (\mathbf{\Pi P}_t \mathbf{\Pi}^\top)^{-1} (\boldsymbol{\sigma} - \mathbf{\Pi x}_t) \right).
$$

Therefore, we have the approximation

$$
\bar{U}(\boldsymbol{\sigma}_{i,j}) \approx A \sum_{m=-N_x}^{N_x} \sum_{n=-N_y}^{N_y} p_{\mathbf{\Pi} X_t}(\mathbf{s}_{i+m,j+n}). \tag{36}
$$

Observing Figure 14 we note that computing $\bar{U}$ for all $\mathbf{s}_{i,j}$ in the camera field-of-view, requires the evaluation of $p_{\mathbf{\Pi} X_t}$ at points over an $(4N_x+1) \times (4N_y+1)$ grid, and that there are many redundant computations. In fact, we have the following lemma.

Lemma 9.3. *Define the* $(2N_x+1) \times (2N_y+1)$ *matrix*

$$
\bar{U}_S \doteq \begin{pmatrix} \bar{U}(s_{-N_x,-N_y}) & \ldots & \bar{U}(s_{-N_x,N_y}) \\ \vdots & & \vdots \\ \bar{U}(s_{-N_x,N_y}) & \ldots & \bar{U}(s_{N_x,N_y}) \end{pmatrix},
$$

and the $(4N_x+1) \times (4N_y+1)$ *matrix*

$$
\mathbf{\Sigma} = \begin{pmatrix} p_{\mathbf{\Pi} X_t}(\mathbf{s}_{-2N_x,-2N_y}) & \ldots & p_{\mathbf{\Pi} X_t}(\mathbf{s}_{-2N_x,2N_y}) \\ \vdots & & \vdots \\ p_{\mathbf{\Pi} X_t}(\mathbf{s}_{-2N_x,2N_y}) & \ldots & p_{\mathbf{\Pi} X_t}(\mathbf{s}_{2N_x,2N_y}) \end{pmatrix},
$$

and let $\mathbf{1}$ *be the matrix of size* $(2N_x+1) \times (2N_y+1)$ *composed of all ones, then*

$$
\bar{U}_S = A \mathbf{\Sigma} \circledast \mathbf{1}
$$

where $\circledast$ *is the non-zero-padded 2D convolution operator.*

Proof. Non-zero-padding implies that the output of the convolution between an $M_1 \times N_1$ matrix and an $M_2 \times N_2$ matrix where $M_1 > M_2$ and $N_1 > N_2$ is a matrix of size $(N_1 - N_2 + 1) \times (M_1 - M_2 + 1)$, implying that $\bar{U}_S$ will be $(4N_x + 1 - (2N_x + 1) + 1) \times (4N_y + 1 - (2N_y + 1) + 1) = (2N_x + 1) \times (2N_y + 1)$. Therefore for $i = -N_x, \ldots, N_x$ and $j = -N_y, \ldots, N_y$, the $(i,j)^{th}$ element of $\bar{U}_S$ is given by

$$\begin{aligned}(\bar{U}_S)_{i,j} &= A \sum_{m=-N_x}^{N_x} \sum_{n=N_y}^{N_y} \mathbf{1}_{-m,-n} \mathbf{\Sigma}_{i+m,j+n} \\ &= A \sum_{m=-N_x}^{N_x} \sum_{n=N_y}^{N_y} p_{\mathbf{\Pi} X_t}(\mathbf{s}_{i+m,j+n})\end{aligned}$$

which is identical to Equation (36). □

10 Conclusions

References

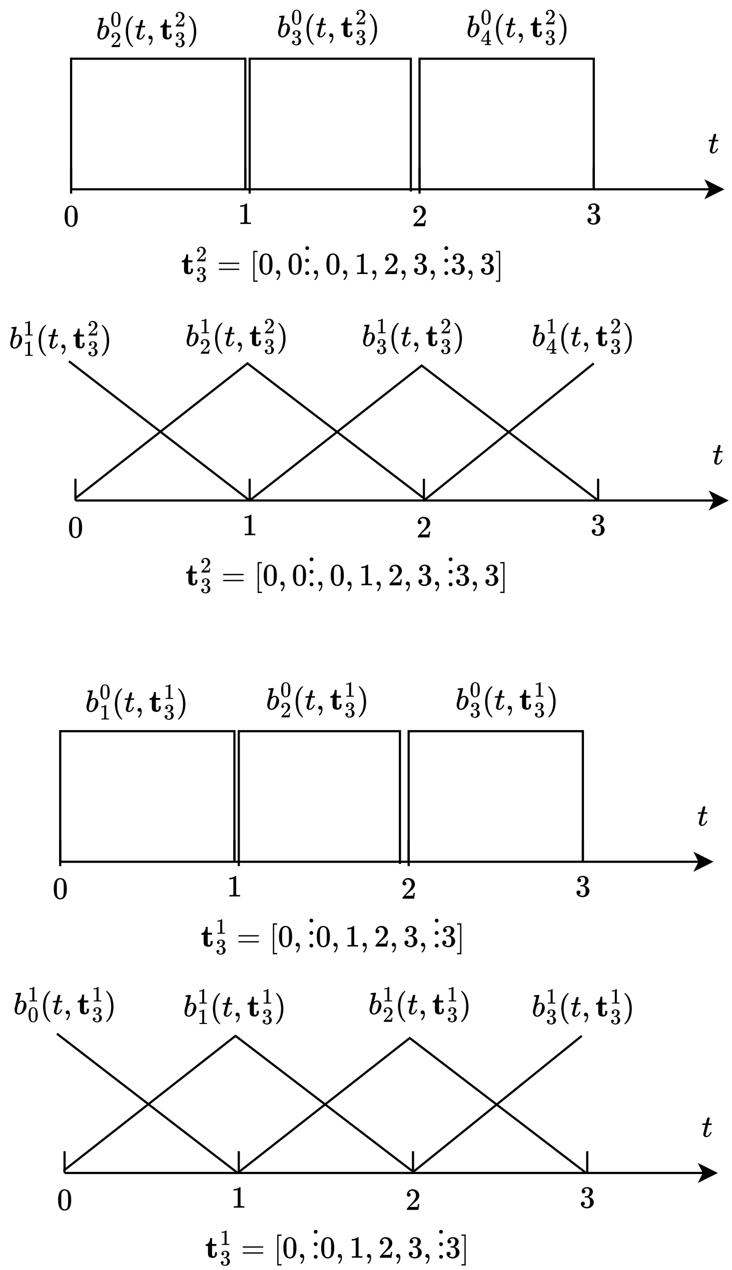

Figure 10: Shifting property of uniform clamped B-spline basis functions.

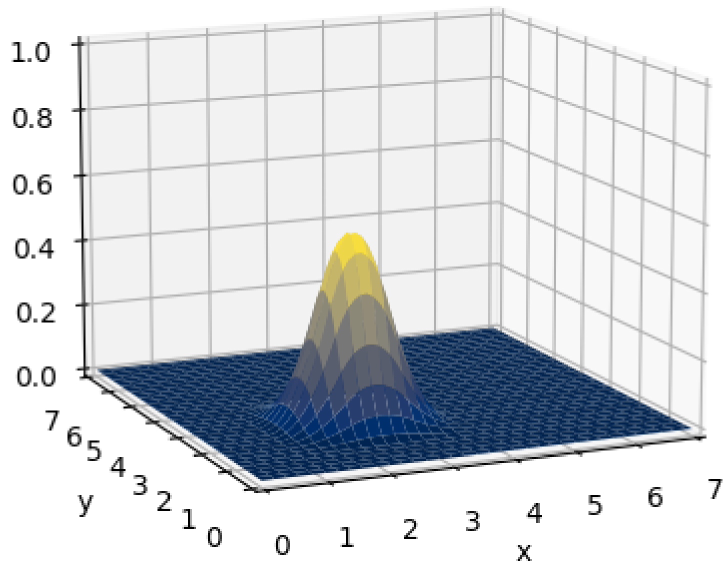

Figure 11: A single second order spline bases function $b_3^2(s_1)b_3^2(s_2)$.

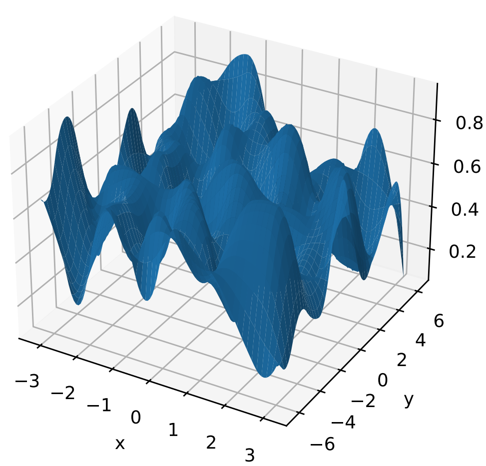

Figure 12: Spline surface with randomly generated control points.

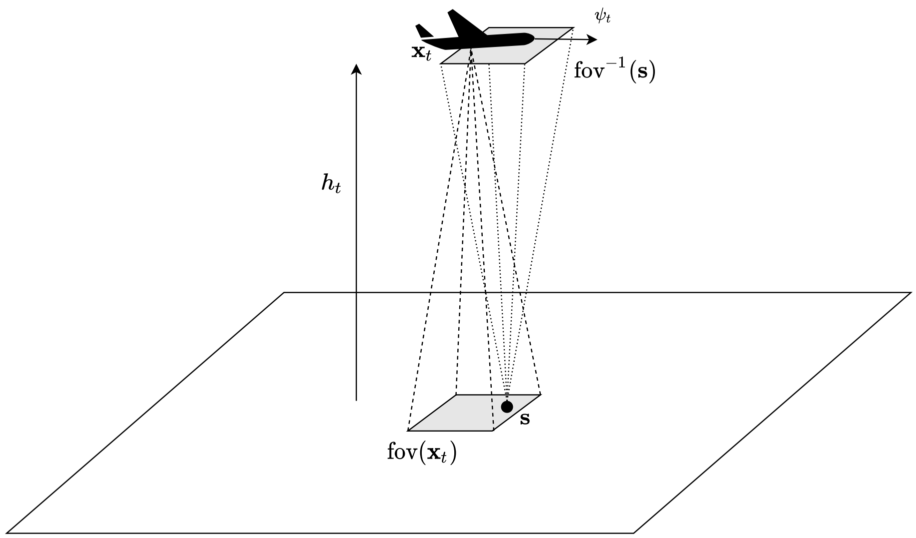

Figure 13: Field-of-view of the camera given $\mathbf{x}_t$, and inverse field-of-view of the camera given $\mathbf{s}$.

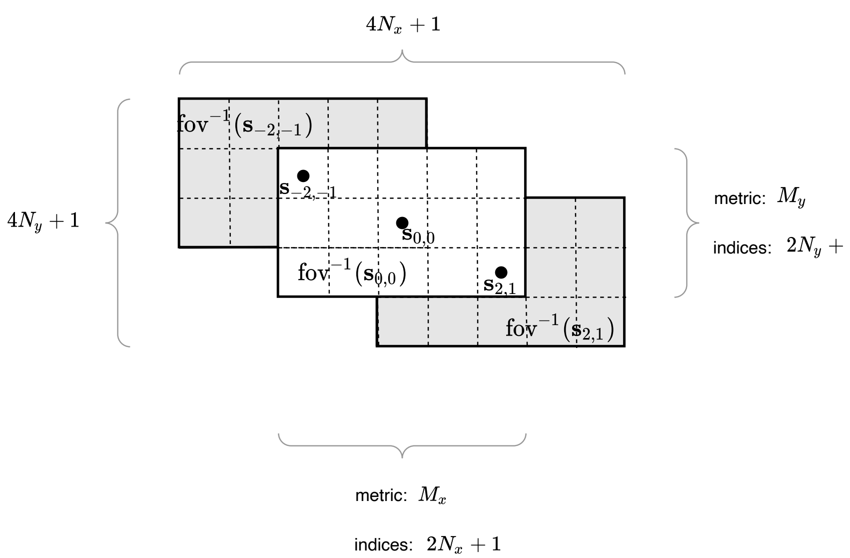

Figure 14: Riemann sum over camera field-of-view.