**CSE 483: Mobile Robotics**

**Lecture by**: Prof. K. Madhava Krishna
**Scribe**:Gourav Kumar, Enna Sachdeva

**Module** # 13
**Date**: 25th October, 2016 (Tuesday)

**Non-holonomic Trajectory Planning (Bernstein Basis method)**

This document discusses the theory of non holonomic trajectory planning using Bernestein polynomial along with the brief description of the associated topics.

$q(t) = \sum_{i=0}^{n} a_i t_i^n$

# 1 What is a Nonholonomic motion

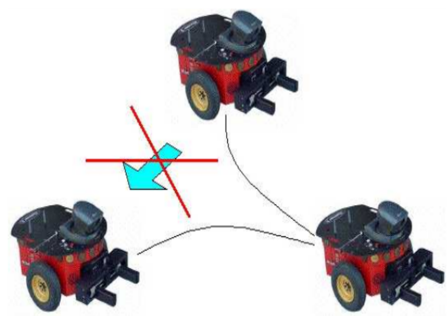


Figure 1:
Non holonomic characteristic of wheeled robots.

Non-holonomic systems are characterized by constraint equations which involves the time derivatives of the system configuration variables. In a configuration space $Q \subset R^n$, the configuration of a mechanical system can be uniquely described by an n-dimensional vector of generalized coordinates.
$q = (q_1, q_2, q_3, .........., q_n)^T$
The generalized velocity at a generic point of a trajectory $q(t) \subset Q$ is the tangent vector given by
$\dot{q} = (\dot{q_1}, \dot{q_2}, \dot{q_3}, .........., \dot{q_n})^T$
For a non holonomic systems, these equations are non integrable as they typically arise when the system has less controls than configuration variables(underactuated systems). As a result, a nonholonomic mechanical system cannot move in arbitrary directions in its configuration space. For instance, a unicycle has two controls (linear($v$) and angular($w$) velocities), while it moves in a 3-dimensional configuration space($x, y, \theta$). As a consequence, any path in the configuration space does not necessarily correspond to a feasible path for the system. In other words, for a non-holonomic systems, the line integrals depend not just on the start and end points but also the path taken.
The state transition matrix representation of a holonomic system is of the form

$$\dot{x} = f(x, u) \tag{1}$$

Since, equation 1 is non-integrable, we can approximate the integration using numerical integration methods, say Euler's method, which gives
$x_{new} \approx x + f(x,u)\Delta t$,
This shows that the new state $x_{new}$ is constrained due to the choice of $f$.
Figure 1 shows one of the feasible paths (represented with lines) of a non-holonomic mobile robot to move between 2 states.

## 2 Differential Drive Robots

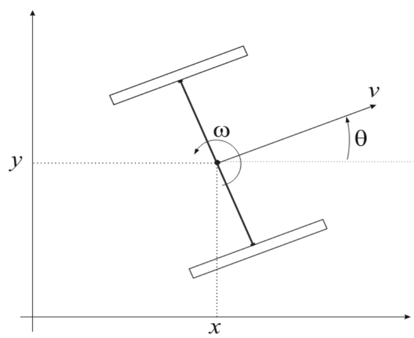



Figure 2:
Differential drive wheeled robot

Consider a differential drive nonholonomic mobile robot in a two-dimensional, free-space environment, as shown in figure 2. It is assumed that the robot cannot slip in lateral direction,
generalized coordinates : $q = (x, y, \theta)^T$
Nonholonomic constraints : $\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \implies \dot{y} = \dot{x}\tan\theta$

$$y = \int \dot{x}\tan\theta dt \tag{2}$$

With 2 control inputs as $(v, w)$, the kinematics model of the system is given by:-

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{3}$$

## 3 Motion Planning with Bernstein Polynomials

Since, equation 2 is non integrable, we can approximate the functions $\dot{x}$ and $\tan\theta$ with the bernestein polynomials and solve the integral.

NOTE: The Bernstein polynomials are advantageous over other approximation techniques like taylor series as the former and its derivatives polynomials uniformly approximates $f$ and $\dot{f}$,respectively. It holds true for higher order derivatives as well. Moreover, they are the most numerically stable basis.

### 3.1 Bernstein basis polynomial

Let $f(x)$ be a real-valued function defined and bounded on the interval [0,1], then $B_n(f)$ is the polynomial on [0,1].

$$\mathbf{B_n(f(x);t) = \sum_{i=0}^{n} f(\frac{i}{n})^n C_i t^i (1-t)^{n-i}} \tag{4}$$

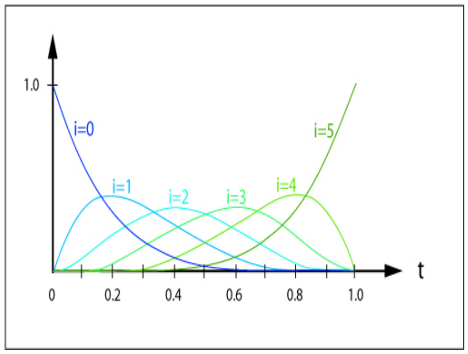



Figure 3:
Bernstein basis polynomials for n=5 .

If function f(x) is continuous on [0,1],then the Bernstein polynomial $B_n(f(x))$ tends uniformly to f as $n \to \infty$. Bernstein basis polynomials with n=5, are shown in figure 3.

For the given initial state $(x_{t_0}, y_{t_0}, \theta_{t_0})$ and final state $(x_{t_f}, y_{t_f}, \theta_{t_f})$,(more state constraints can be added to the system), with the start time $t_0$ and end time $t_f$ of the trajectory of a nonholonomic system, $\dot{y(t)}$ and $\dot{x(t)}$ can be related as,
$\dot{y(t)} = \dot{x(t)} \tan\theta(t)$,
where, the functions $x(t)$ and $\tan\theta(t)$ can be approximated as a linear combination of Bernstein basis polynomials as following-

$$x(t) \approx B_n(x(t)) = B_x(\mu(t)) = \sum_{i=0}^{5} W_{x_i} B_i(\mu(t)), \tag{5}$$

Similarly,

$$\tan\theta(t) = k(t) \approx B_n(k(t)) = B_k(\mu(t)) = \sum_{i=0}^{5} W_{k_i} B_i(\mu(t)) \tag{6}$$

where,
$B_i(\mu(t)) =^n C_i(1-\mu)^i(\mu)^{n-i}$
$\mu(t) = \frac{t-t_0}{t_f-t_0}$
Differentiating equation 5 w.r.t time gives

$$\dot{x}(t) = \dot{B}_x(\mu(t)) = \sum_{i=0}^{5} W_{x_i} \dot{B}_i(\mu(t) \tag{7}$$

using above 2 equations, equation 2 can be rewritten as-

$$y(t) = y_0 + \int_{t_0}^{t} (\sum_{i=0}^{5} W_{x_i} \dot{B}_i(\mu(t))(\sum_{i=0}^{5} W_{k_i} B_i(\mu(t))dt \tag{8}$$

The bernstein coefficients of the polynomials and their derivatives for n=5, at time $t = t_0$ and $t = t_f$ are shown in the tables below.

| Bernstein coefficients | $t = t_0$ | $t = t_f$ |
|---|---|---|
| $B_0(\mu) =^5 C_0(1-\mu)^5\mu^0$ | 1 | 0 |
| $B_1(\mu) =^5 C_1(1-\mu)^4\mu$ | 0 | 0 |
| $B_2(\mu) =^5 C_2(1-\mu)^3\mu^2$ | 0 | 0 |
| $B_3(\mu) =^5 C_3(1-\mu)^2\mu^3$ | 0 | 0 |
| $B_4(\mu) =^5 C_4(1-\mu)^1\mu^4$ | 0 | 0 |
| $B_5(\mu) =^5 C_5(1-\mu)^0\mu^5$ | 0 | 1 |

| Bernstein coefficients derivatives | $t = t_0, \mu = 0$ | $t = t_f, \mu = 1$ |
|---|---|---|
| $\dot{B}_0(\mu) =^5 C_0 \frac{-5(1-\mu)^4}{(t_f-t_0)}$ | $\frac{-5}{(t_f-t_0)}$ | 0 |
| $\dot{B}_1(\mu) =^5 C_1 \frac{-4\mu(1-\mu)^3+(1-\mu)^4}{(t_f-t_0)}$ | $\frac{5}{(t_f-t_0)}$ | 0 |
| $\dot{B}_2(\mu) =^5 C_2 \frac{-3\mu^2(1-\mu)^2+2(1-\mu)^3\mu}{(t_f-t_0)}$ | 0 | 0 |
| $\dot{B}_3(\mu) =^5 C_3 \frac{-2\mu^3(1-\mu)+3(1-\mu)^2\mu^2}{(t_f-t_0)}$ | 0 | 0 |
| $\dot{B}_4(\mu) =^5 C_4 \frac{-\mu^4+4(1-\mu)\mu^3}{(t_f-t_0)}$ | 0 | $\frac{-5}{(t_f-t_0)}$ |
| $\dot{B}_5(\mu) =^5 C_5 \frac{5\mu^4}{(t_f-t_0)}$ | 0 | $\frac{5}{(t_f-t_0)}$ |

With the given state constraints(i.e. position, velocity, acceleration, etc.) of the robot at different instants, the unknown, time independent weight parameters $(W_{x_0}, W_{x_1}, W_{x_2}, ...., W_{x_5})$ and $(W_{k_0}, W_{k_1}, W_{k_2}, ...., W_{k_5})$ can be determined.

## 3.2 Finding $W_{x_0}, W_{x_1}, ...., W_{x_5}$

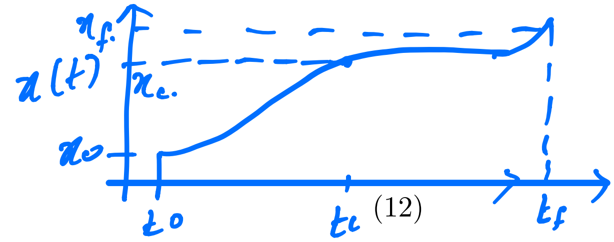
Given Constraints : $(x_{t_0}, y_{t_0}), (x_{t_c}, y_{t_c}), (x_{t_f}, y_{t_f}), (\dot{x_{t_0}}, \dot{y_{t_0}}), (\dot{x_{t_c}}, \dot{y_{t_c}}), (\dot{x_{t_f}}, \dot{y_{t_f}})$
Using equations 5, known parameters can be represented as a linear combination of bernstein basis polynomial as follows-

$$x(t_0) = W_{x_0}B_0(\mu(t_0)) + W_{x_1}B_1(\mu(t_0)) + W_{x_2}B_2(\mu(t_0)) + W_{x_3}B_3(\mu(t_0)) + W_{x_4}B_4(\mu(t_0)) + W_{x_5}B_5(\mu(t_0)) \tag{9}$$

$$x(t_f) = W_{x_0}B_0(\mu(t_f)) + W_{x_1}B_1(\mu(t_f)) + W_{x_2}B_2(\mu(t_f)) + W_{x_3}B_3(\mu(t_f)) + W_{x_4}B_4(\mu(t_f)) + W_{x_5}B_5(\mu(t_f)) \tag{10}$$

Putting values of bernstein polynomial coefficients (from tables) in the equations 9 and 10, gives

$$W_{x_0} = x(t_0) = x_{t_0} \tag{11}$$




$$W_{x_5} = x(t_f) = x_{t_f} \tag{12}$$

Using the remaining constraints, all weights $W_{x_1}, W_{x_2}, W_{x_3}, W_{x_4}$ can be evaluated.

$$\begin{bmatrix} x_{t_c} - W_{x0}B_0(\mu(t_c)) - W_{x_5}B_5(\mu(t_c)) \\ \dot{x}_{t_0} - W_{x0}\dot{B}_0(\mu(t_0)) - W_{x_5}\dot{B}_5(\mu(t_0)) \\ \dot{x}_{t_f} - W_{x0}\dot{B}_0(\mu(t_f)) - W_{x_5}\dot{B}_5(\mu(t_f)) \\ \dot{x}_{t_c} - W_{x0}\dot{B}_0(\mu(t_c)) - W_{x_5}\dot{B}_5(\mu(t_c)) \end{bmatrix} = \begin{bmatrix} B_1(\mu(t_c)) & B_2(\mu(t_c)) & B_3(\mu(t_c)) & B_4(\mu(t_c)) \\ \dot{B}_1(\mu(t_0)) & \dot{B}_2(\mu(t_0)) & \dot{B}_3(\mu(t_0)) & \dot{B}_4(\mu(t_0)) \\ \dot{B}_1(\mu(t_f)) & \dot{B}_2(\mu(t_f)) & \dot{B}_3(\mu(t_f)) & \dot{B}_4(\mu(t_f)) \\ \dot{B}_1(\mu(t_c)) & \dot{B}_2(\mu(t_c)) & \dot{B}_3(\mu(t_c)) & \dot{B}_4(\mu(t_c)) \end{bmatrix} \begin{bmatrix} W_{x_1} \\ W_{x_2} \\ W_{x_3} \\ W_{x_4} \end{bmatrix} \tag{13}$$

$$A_x = B_x W_x \tag{14}$$

$$W_x = pinv(B_x)A_x \tag{15}$$

### 3.3 Finding $W_{k_0}, W_{k_1}, ...., W_{k_5}$

Expanding equation 8,

$$y(t) = y_0 + \int_{t_0}^{t} (W_{k_0}.f_0(t, t_0, t_f, W_{x_1}, W_{x_2}, ...W_{x_5}) + (W_{k_1}.f_1(t, t_0, t_f, W_{x_1}, W_{x_2}, ...W_{x_5}) + ..... + (W_{k_5}.f_5(t, t_0, t_f, W_{x_1} \tag{16}$$

with the weight parameters $W_{x_1}....W_{x_5}$ calculated above, equation 16 further reduces to,

$$y(t) = y_0 + W_{k_0}F_0(t) + W_{k_1}F_1(t) + W_{k_2}F_2(t) + W_{k_3}F_3(t) + W_{k_4}F_4(t)0 + W_{k_5}F_5(t) \tag{17}$$

$$where F_i(t) = \int_{t_0}^{t} f_i(t, t_0, t_f, W_{x_1}, W_{x_2}, ...W_{x_5})dt \tag{18}$$

is a polynomial & so is $\int f_i(\cdot)$

However (18) is tedious to evaluate. The values are stored & kept

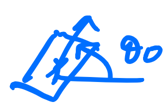
Our objective is to get weights, $W_{K_0}, W_{K_1}, W_{K_2}, W_{K_3}, W_{K_4}, W_{K_5}$

Typically use Mathematica to get (18).

$$k(t_0) = k_0 = W_{k_0}B_0(\mu(t_0)) + W_{k_1}B_1(\mu(t_0)) + W_{k_2}B_2(\mu(t_0)) + W_{k_3}B_3(\mu(t_0)) + W_{k_4}B_4(\mu(t_0)) + W_{k_5}B_5(\mu(t_0)) \tag{19}$$

which gives,

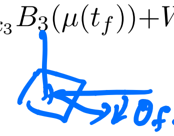
$$W_{k_0} = k_0 \tag{20}$$




$$k(t_f) = k_f = W_{k_0}B_0(\mu(t_f)) + W_{k_1}B_1(\mu(t_f)) + W_{k_2}B_2(\mu(t_f)) + W_{k_3}B_3(\mu(t_f)) + W_{k_4}B_4(\mu(t_f)) + W_{k_5}B_5(\mu(t_f)) \tag{21}$$

which gives,

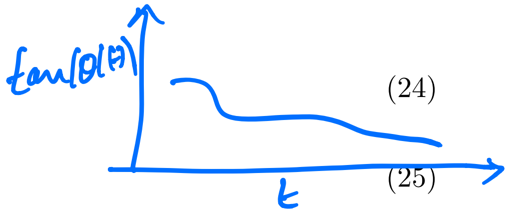
$$W_{k_5} = k_f \tag{22}$$




Using any 4 out of remaining constraints on y and k we can form a full rank matrix for $B_k$ as:

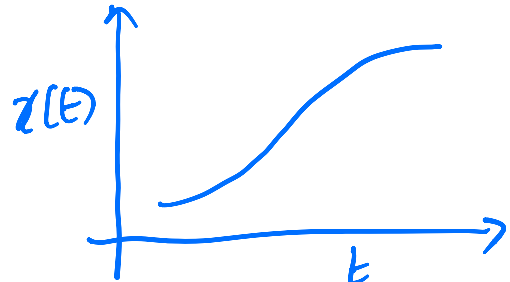
$$\begin{bmatrix} \dot{k}_{t_0} - W_{k_0}\dot{B}_0(\mu(t_0)) - W_{k_5}\dot{B}_5(\mu(t_0)) \\ \dot{k}_{t_f} - W_{k_0}\dot{B}_0(\mu(t_f)) - W_{k_5}\dot{B}_5(\mu(t_f)) \\ y_0 - W_{k_0}F_0 - W_{k_5}F_5 \\ y_f - W_{k_5}F_0 - W_{k_5}F_5 \end{bmatrix} = \begin{bmatrix} \dot{B}_{k_1}(\mu(t_0)) & \dot{B}_2(\mu(t_0)) & \dot{B}_3(\mu(t_0)) & \dot{B}_4(\mu(t_0)) \\ \dot{B}_1(\mu(t_f)) & \dot{B}_2(\mu(t_f)) & \dot{B}_3(\mu(t_f)) & \dot{B}_4(\mu(t_f)) \\ F_1(t_0) & F_2(t_0) & F_3(t_0) & F_4(t_0) \\ F_1(t_f) & F_2(t_f) & F_3(t_f) & F_4(t_f) \end{bmatrix} \begin{bmatrix} W_{k_1} \\ W_{k_2} \\ W_{k_3} \\ W_{k_4} \end{bmatrix} \tag{23}$$

$$A_k = B_k W_k \tag{24}$$

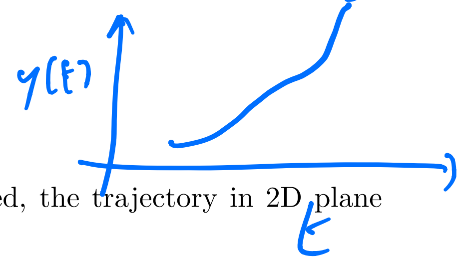



$$W_k = pinv(B_k) A_k \tag{25}$$

Depending on the number of given constraints i.e., rank of the matrices $B_x$ $(p \times q)$ and $B_k$ $(m \times n)$ , the system can be categorized as under constrained, critically constrained and over constrained.




1. **Critically constrained:**
   p=q, rank($B_x$)=$p$ $\implies$ $pinv(B_x) = B_x^{-1}$
   m=n, rank($B_k$)=$m$ $\implies$ $pinv(B_k) = B_k^{-1}$
2. **Under constrained:**
   $p < q$, rank$(B_x) = p$ $\implies$ $pinv(B_x) = (B_x^T B_x)^{-1} B_x^T$
   $m < n$, rank$(B_k) = m$ $\implies$ $pinv(B_k) = (B_k^T B_k)^{-1} B_k^T$
3. **Over constrained:**
   $p > q$, rank$(B_x) = q$ $\implies$ $pinv(B_x) = B_x^T (B_x B_x^T)^{-1}$
   $m > n$, rank$(B_k) = n$ $\implies$ $pinv(B_k) = B_k^T (B_k B_k^T)^{-1}$




Once, the weight parameters $W_k$ and $W_x$ are determined, the trajectory in 2D plane can be estimated as following,

$$x(t) = W_{x_0} B_0(\mu(t)) + W_{x_1} B_1(\mu(t)) + W_{x_2} B_2(\mu(t)) + W_{x_3} B_3(\mu(t)) + W_{x_4} B_4(\mu(t)) + W_{x_5} B_5(\mu(t)) \tag{26}$$

$$y(t) = y_0 + W_{k_0} F_0(t) + W_{k_1} F_1(t) + W_{k_2} F_2(t) + W_{k_3} F_3(t) + W_{k_4} F_4(t) + W_{k_5} F_5(t) \tag{27}$$

and the orientation of the robot can be defined by-

$$\theta(t) = \arctan((W_{k_0} B_0(\mu(t) + W_{k_1} B_1(\mu(t)) + W_{k_2} B_2(\mu(t)) + W_{k_3} B_3(\mu(t)) + W_{k_4} B_4(\mu(t)) + W_{k_5} B_5(\mu(t)) \tag{28}$$