Università di Roma
Tor Vergata

UNIVERSITÀ DEGLI STUDI DI ROMA
“TOR VERGATA”

Dipartimento di Ingegneria dell’Impresa

Implementazione numerica di Spline Classiche Polinomiali, B-Spline e NURBS

Corsi di:
Prototipazione virtuale
Prototipazione virtuale e simulazione dei sistemi meccanici

Ing. Marco Cirelli

24/10/2018

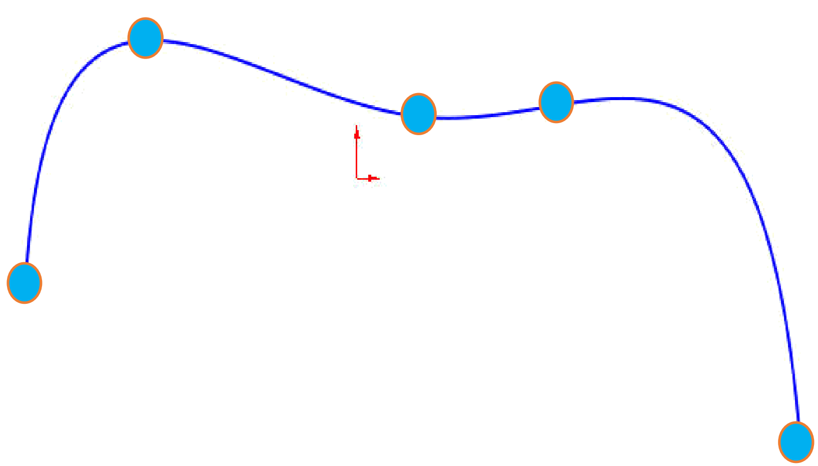
Tipologia di curve

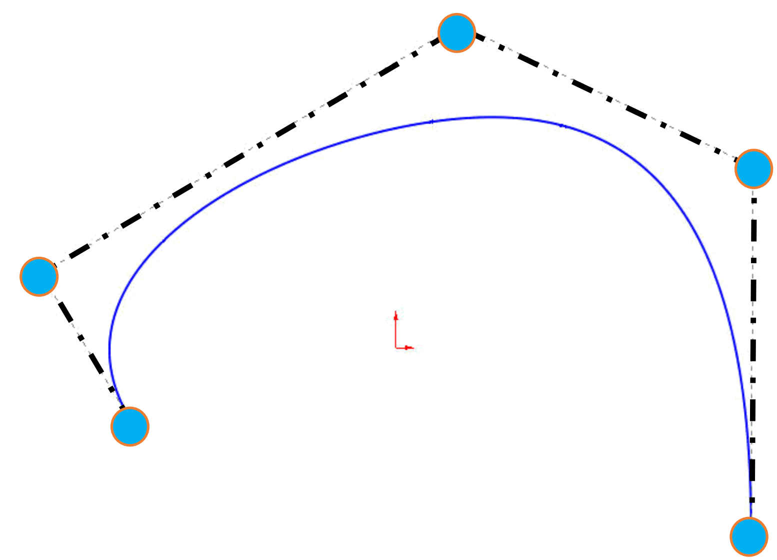
Interpolanti

Dato un insieme di punti trovare la struttura che passi per ogni punto con particolari requisiti di continuità

Approssimanti

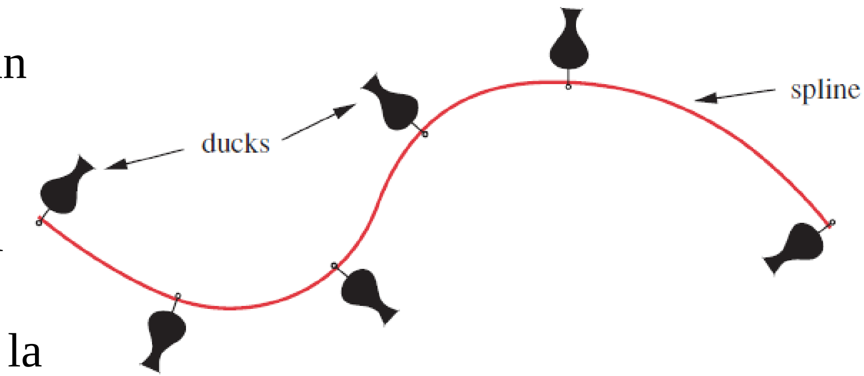
Dato un insieme di punti trovare la struttura la cui forma approssimi l’insieme di punti (estetica)

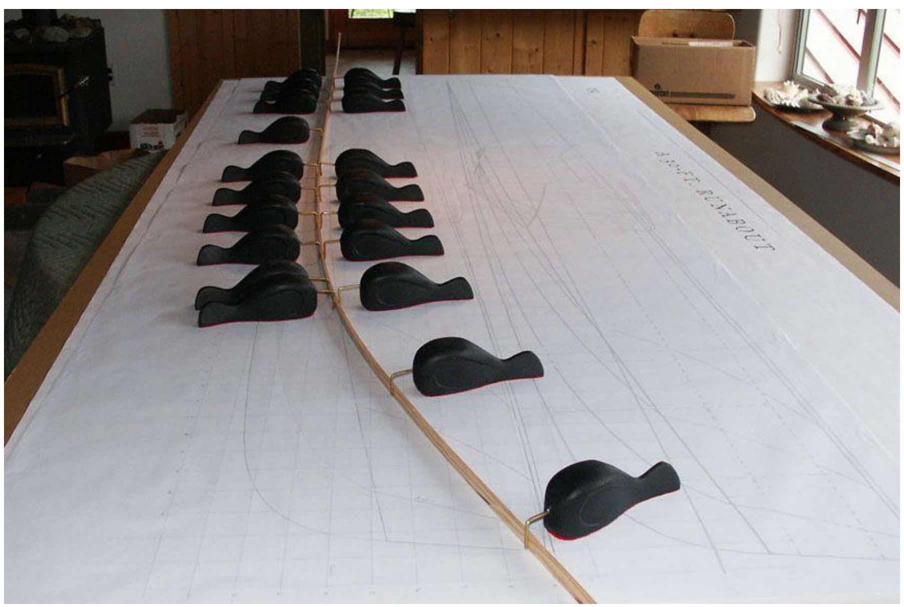




Concetto di Spline

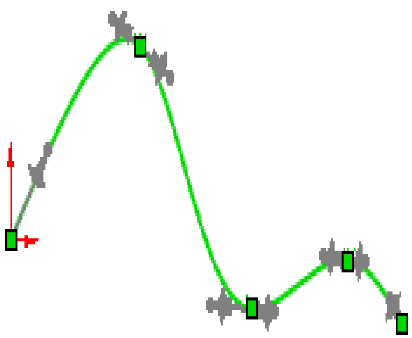
Quando non esistevano i sistemi CAD, i disegnatori utilizzavano delle aste flessibili in plastica o bamboo per disegnare delle curve interpolanti. Tali curve che assumevano la minima energia di deformazione, soggetta ai vincoli imposti (chiamati «ducks»). Il disegnatore in questo modo poteva tracciare la curva più "smooth" che passasse attraverso i punti.

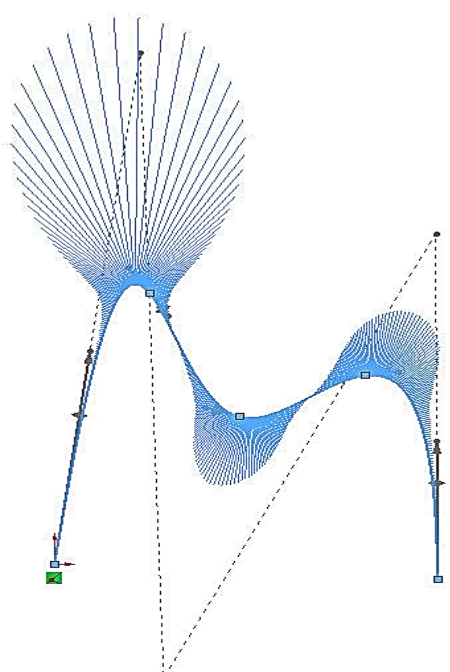


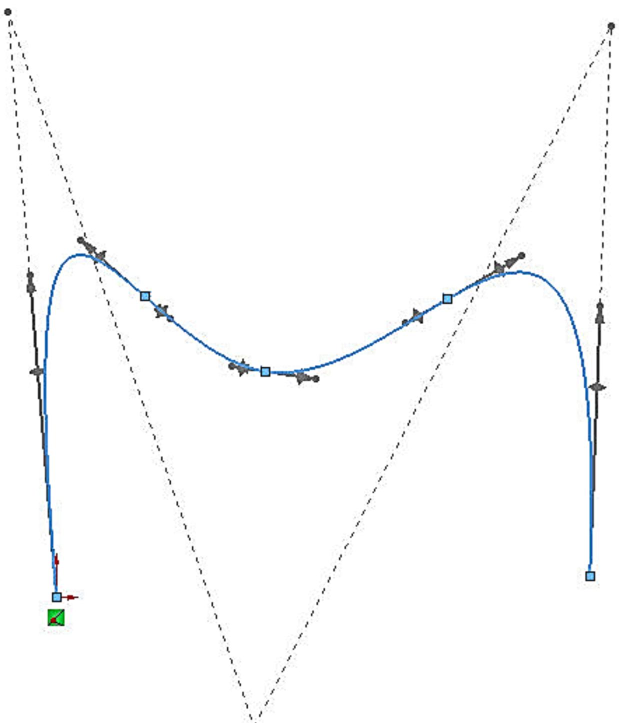


Spline nel Computer Aided Design

Attualmente, attraverso l'utilizzo di software di modellazione, è possibile generare delle spline che seguono dei **modelli matematici** di definizione.
Generalmente si utilizzano **punti di interpolazione per la creazione**, e **punti di controllo per la modifica della geometria** (operazione che rende la modifica più uniformemente distribuita).







Modelli matematici di spline che vedremo

Spline classica polinomiale

Una spline classica polinomiale una curva costruita da diversi polinomi, dello stesso grado che attaccati tra loro.

$$S_j = A_j(\theta-\theta_j)^5 + B_j(\theta-\theta_j)^4 + C_j(\theta-\theta_j)^3 + D_j(\theta-\theta_j)^2 + E_j(\theta-\theta_j) + F_j$$

B-spline

Una B-spline rappresenta una spline costruita utilizzando più curve di Bezier attaccate tra loro.

$$s(x) = \sum_{j=1}^{n} N_{j,k}(x) A_j \qquad T_1 \le x \le T_{k+n}$$

Vettore dei nodi di controllo

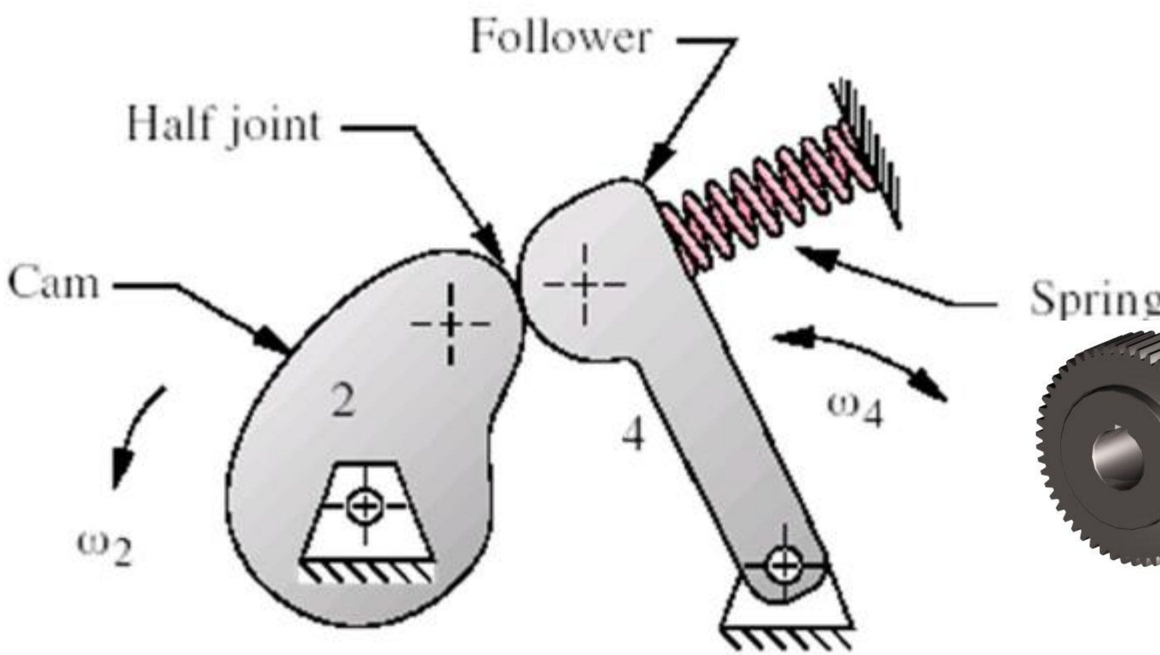
NURBS

Una NURBS (Non Uniform Rational B-spline) rappresenta una B-spline con vettore dei nodi non uniforme e caratterizzate dalla presenza di pesi.

$$s(x) = \sum_{j=1}^{n} R_{j,k}(x) A_j \qquad T_1 \le u \le T_{k+n} \qquad R_{j,k}(x) = \frac{N_{j,k}(x) W_j}{\sum_{j=1}^{n} N_{j,k}(x) W_j}$$

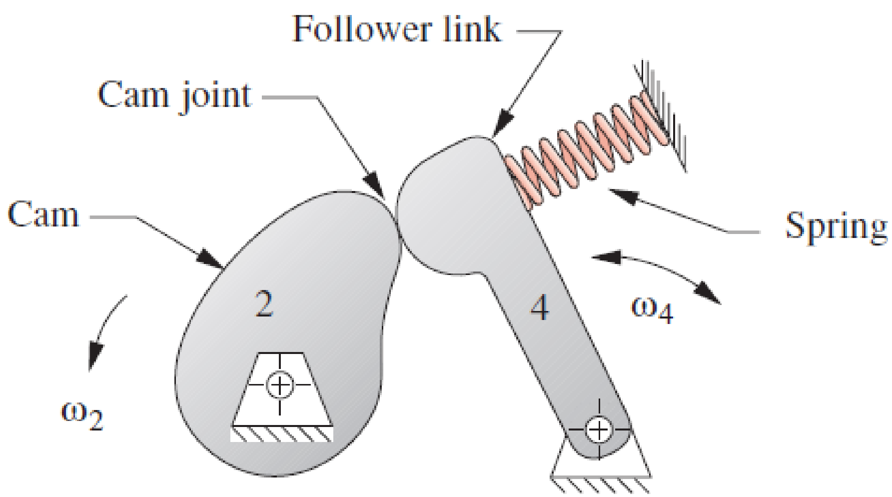
Meccanismi a camma

Si definisce **camma** un **membro sagomato** in modo tale da **«comunicare» un moto** di assegnate caratteristiche cinematiche ad un altro membro, detto **cedente**, con il quale deve essere in **contatto continuo**. Tale continuità, può essere garantita solo se il profilo presenta continuità sulle derivate successive (terza, quarta), poiché altrimenti, le elevate velocità possono creare effetti di distacco.

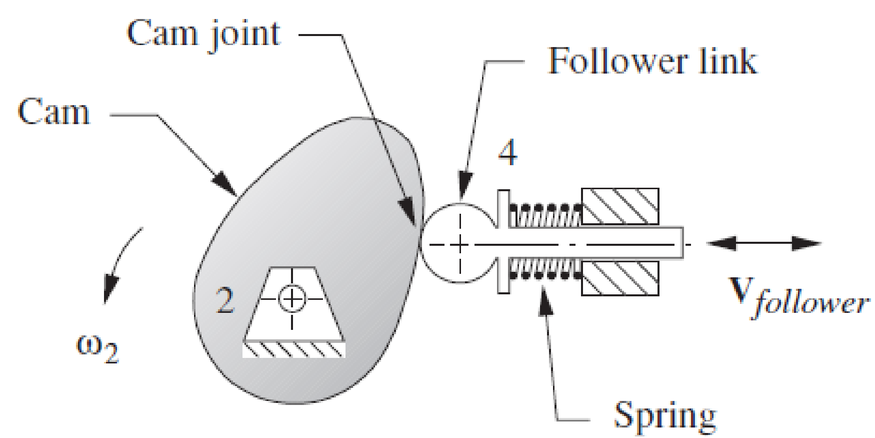


Classificazione del meccanismo sul moto assoluto del cedente

Seconda specie: Moto del cedente oscillatorio

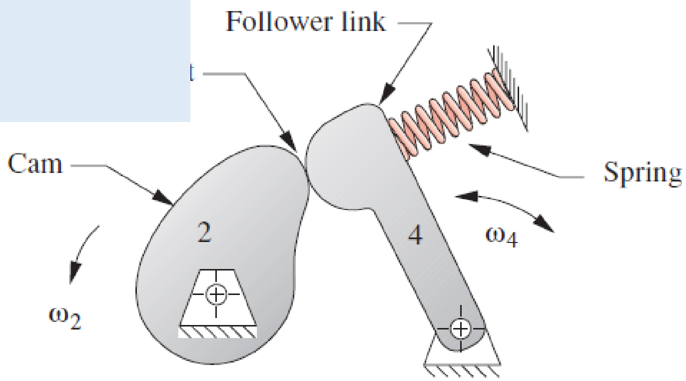


Prima specie: Moto del cedente traslatori

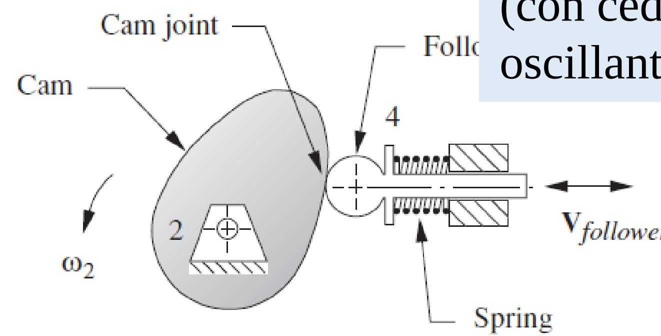


Classificazione del meccanismo per tipo di «chiusura»

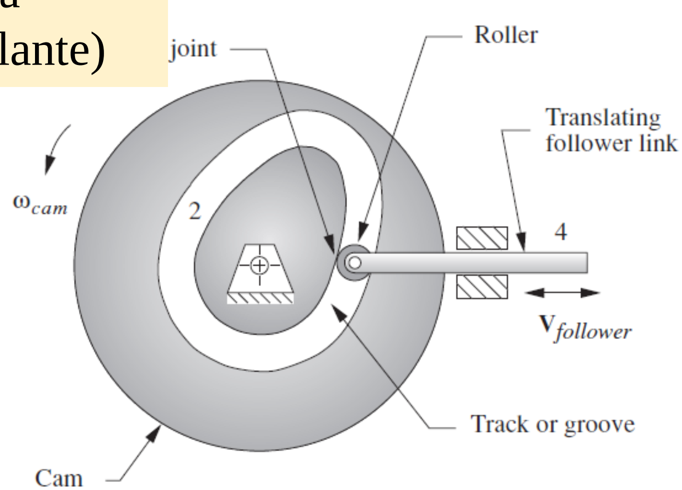
Chiusura di forza
(con cedente
traslante)



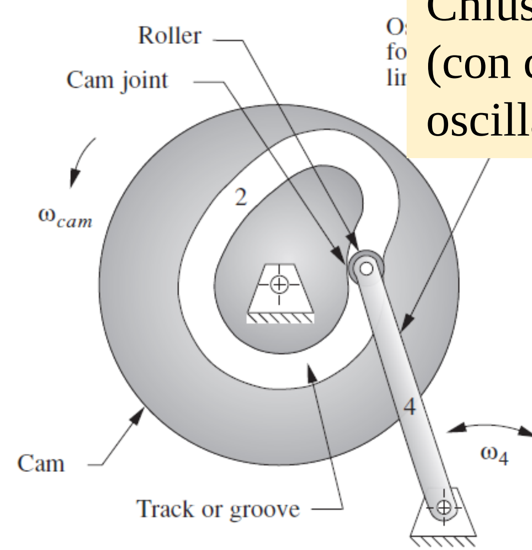
Chiusura di forza
(con cedente
oscillante)



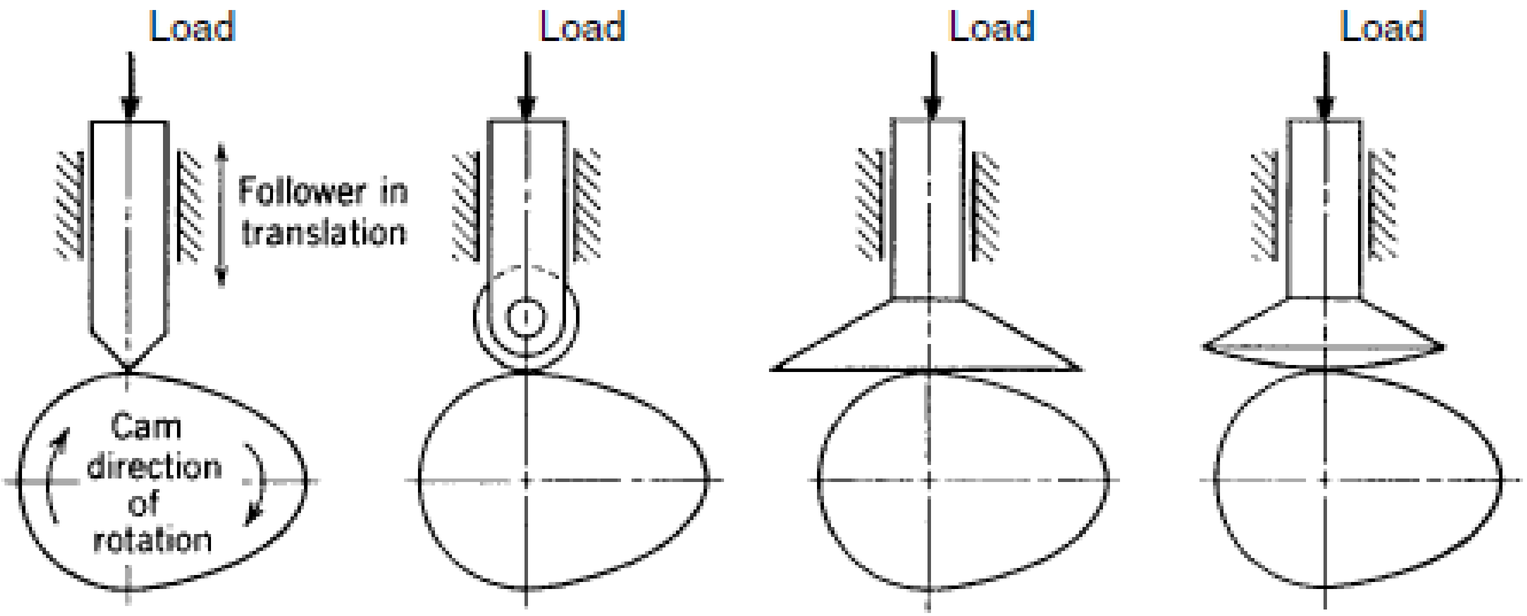
Chiusura di forma
(con cedente traslante)



Chiusura di forma
(con cedente
oscillante)



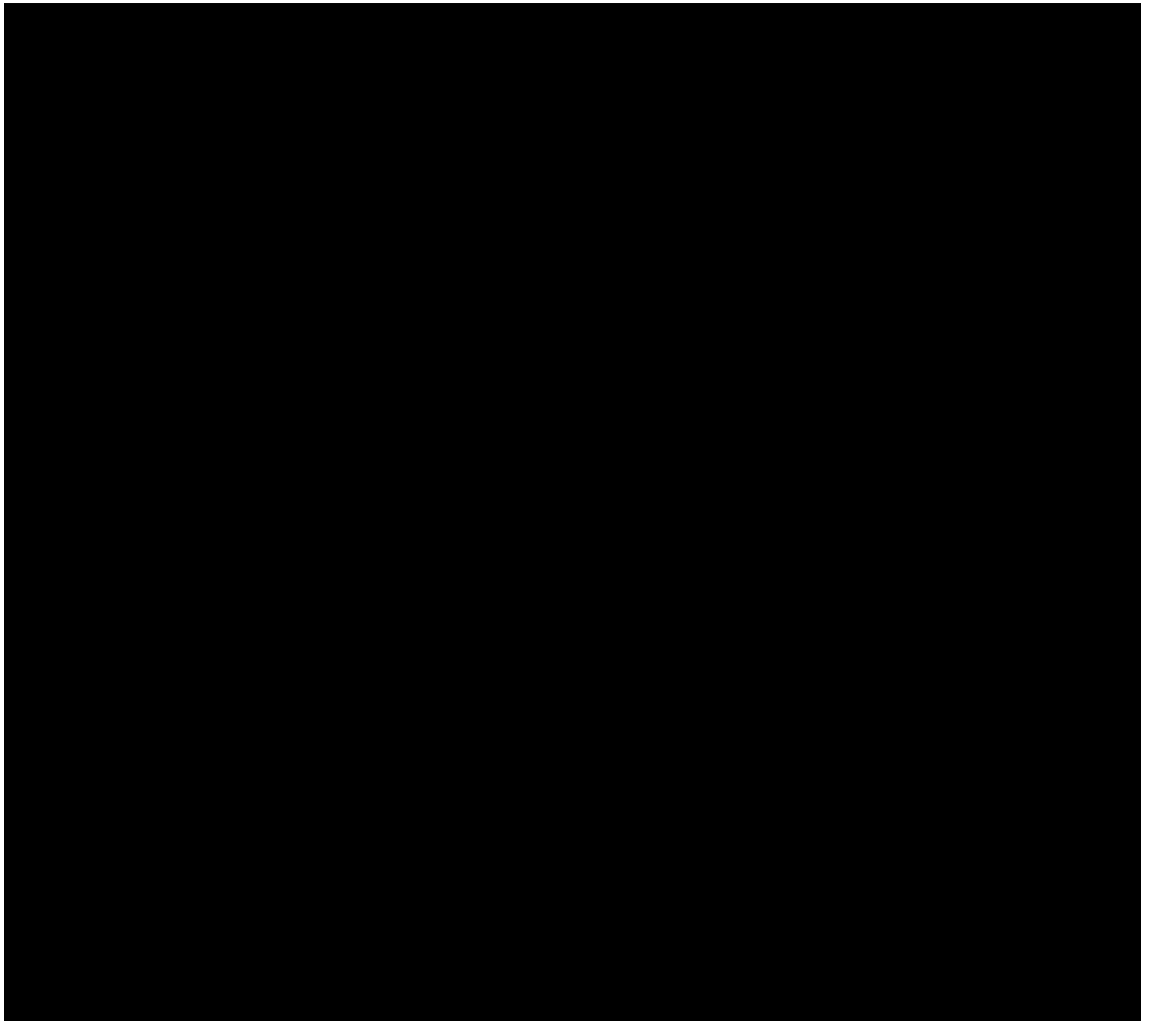
Classificazione del meccanismo per tipo di cedente



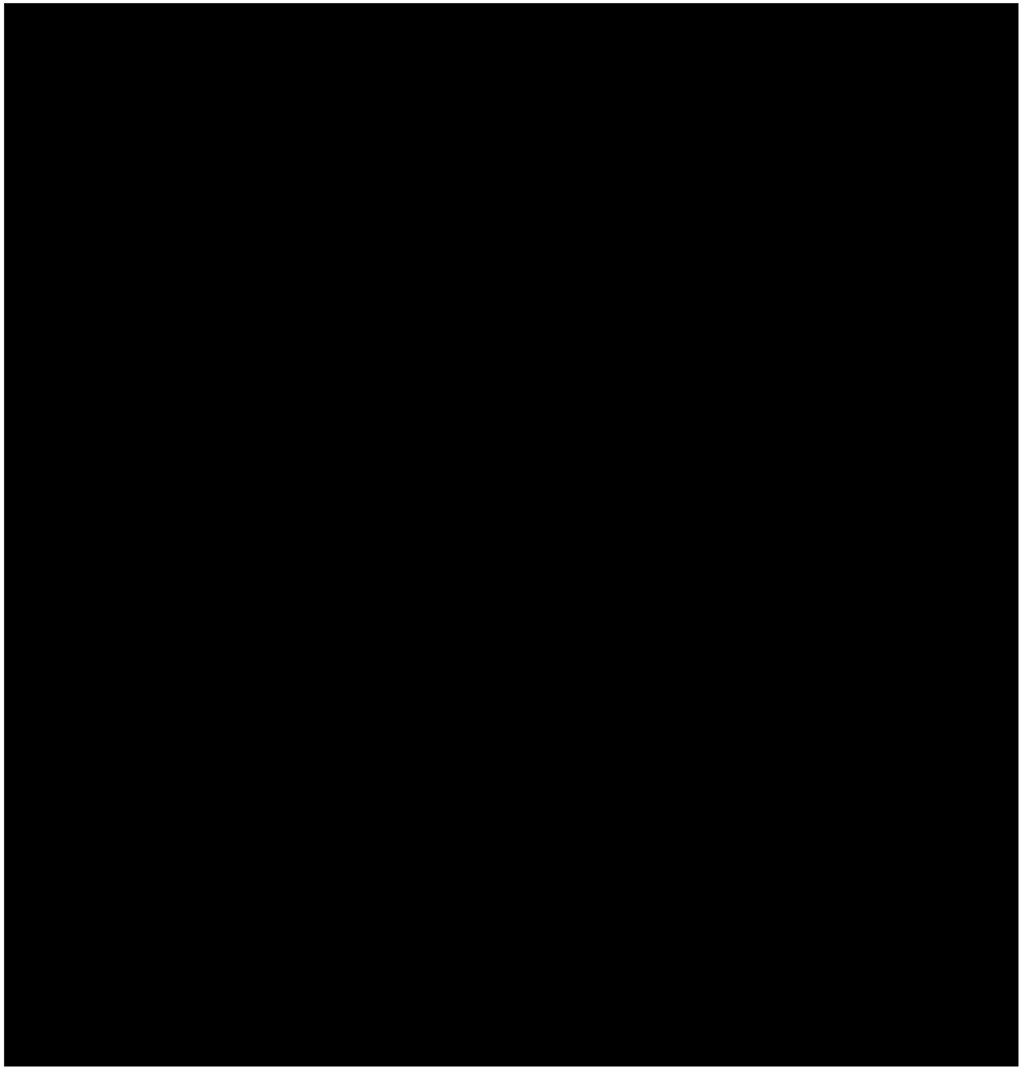
(*a*) Knife-edge follower.
(*b*) Roller follower.
(*c*) Flat-faced follower.
(*d*) Spherical-faced follower.

Classificazione del meccanismo in base al tipo di camma

Esempi di applicazione



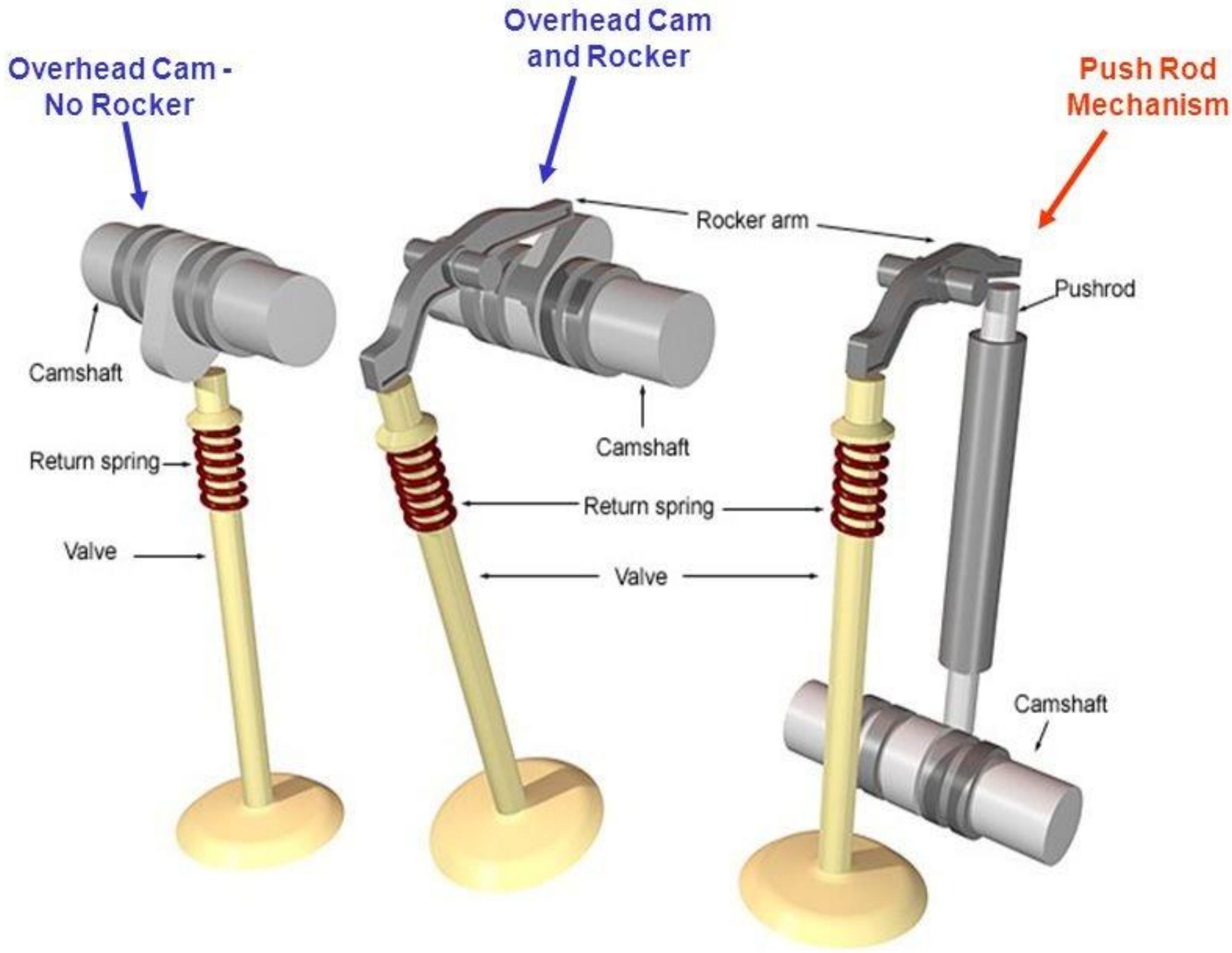
Esempi di applicazione



Applicazione per motori a combustione interna (MCI)

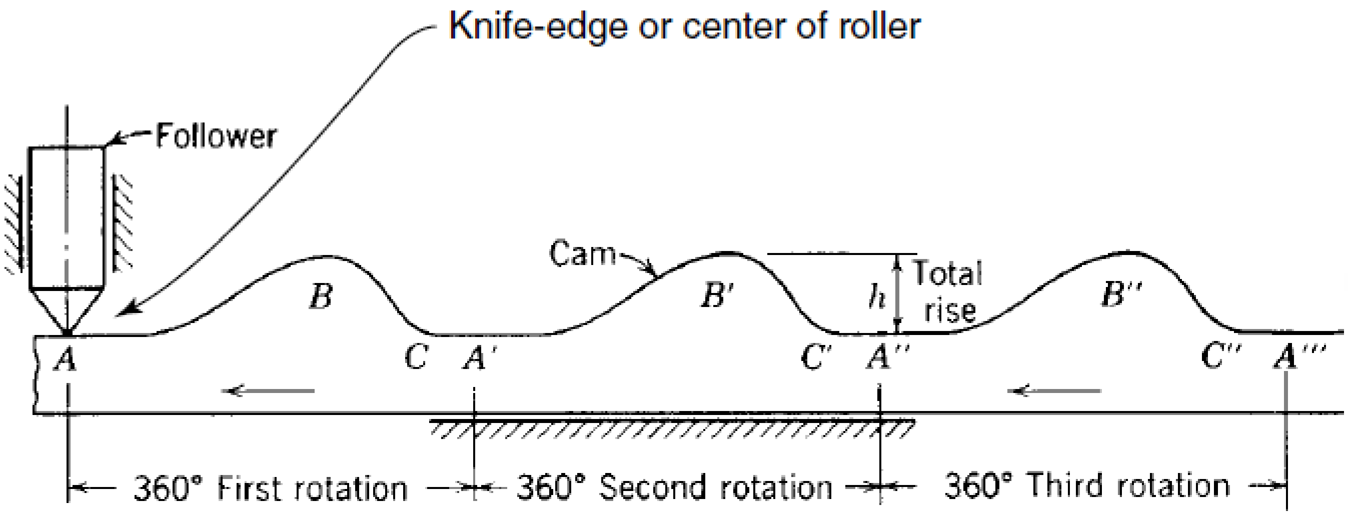
Applicazione per motori a combustione interna (MCI)

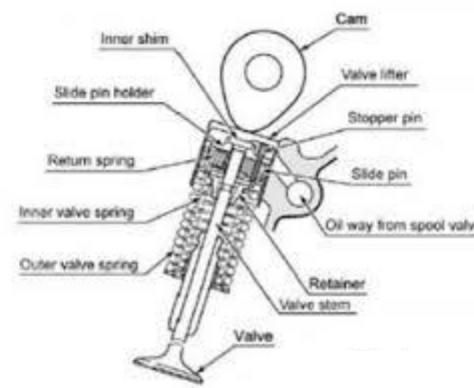
Meccanismi a camma per MCI



Progettazione di una camma

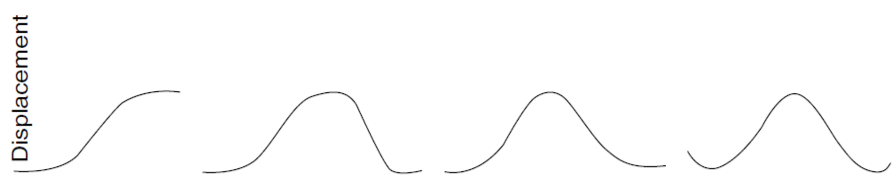
La prima fase del progetto di una camma inizia dalla definizione del *diagramma delle alzate*, in cui viene esplicitata la posizione del cedente, in funzione di quella del movente.





ATTENZIONE: Nelle fasi preliminari del progetto, il profilo delle alzate deve essere sviluppato in maniera sinergica tra il settore dinamico e quello fluidodinamico per garantire il target di prestazione motoristica richiesto.

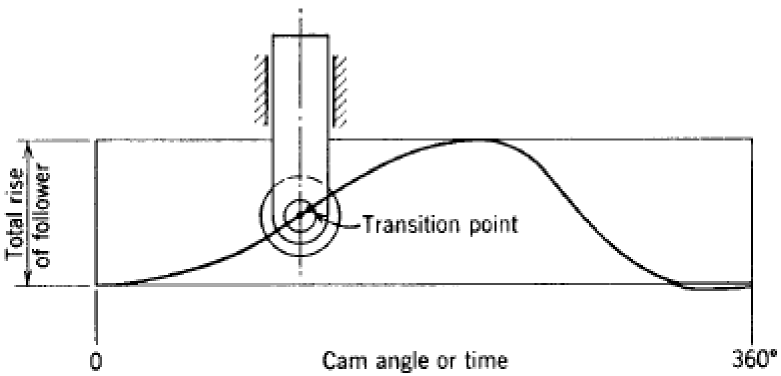
Tipologia di alzata studiata in fase preliminare:

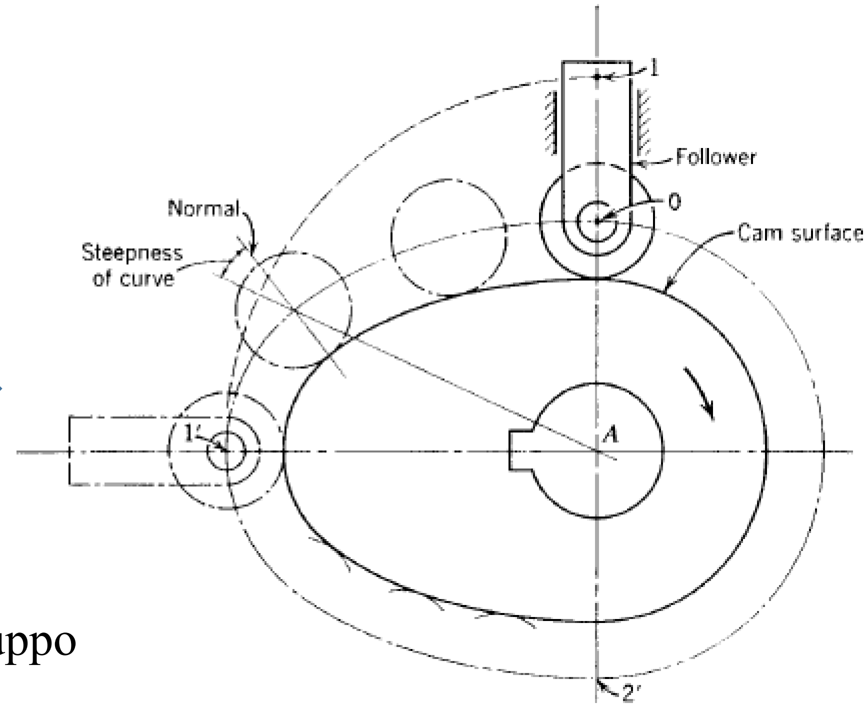


(*a*) Dwell-rise-dwell (DRD). (*b*) Dwell-rise-dwell-return-dwell (DRDRD). (*c*) Dwell-rise-return-dwell (DRRD). (*d*) Rise-retrun-rise (RRR).

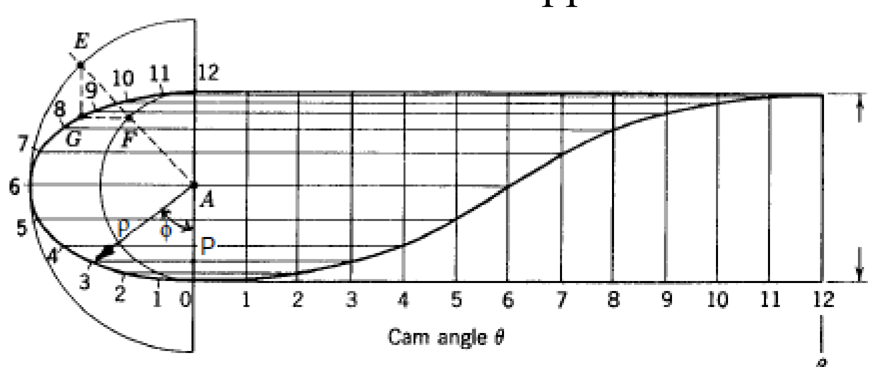
Progettazione di una camma

La seconda fase, avvalendosi di costruzioni geometriche, si procede con la definizione della sagoma della camma a partire dal diagramma delle alzate definito.



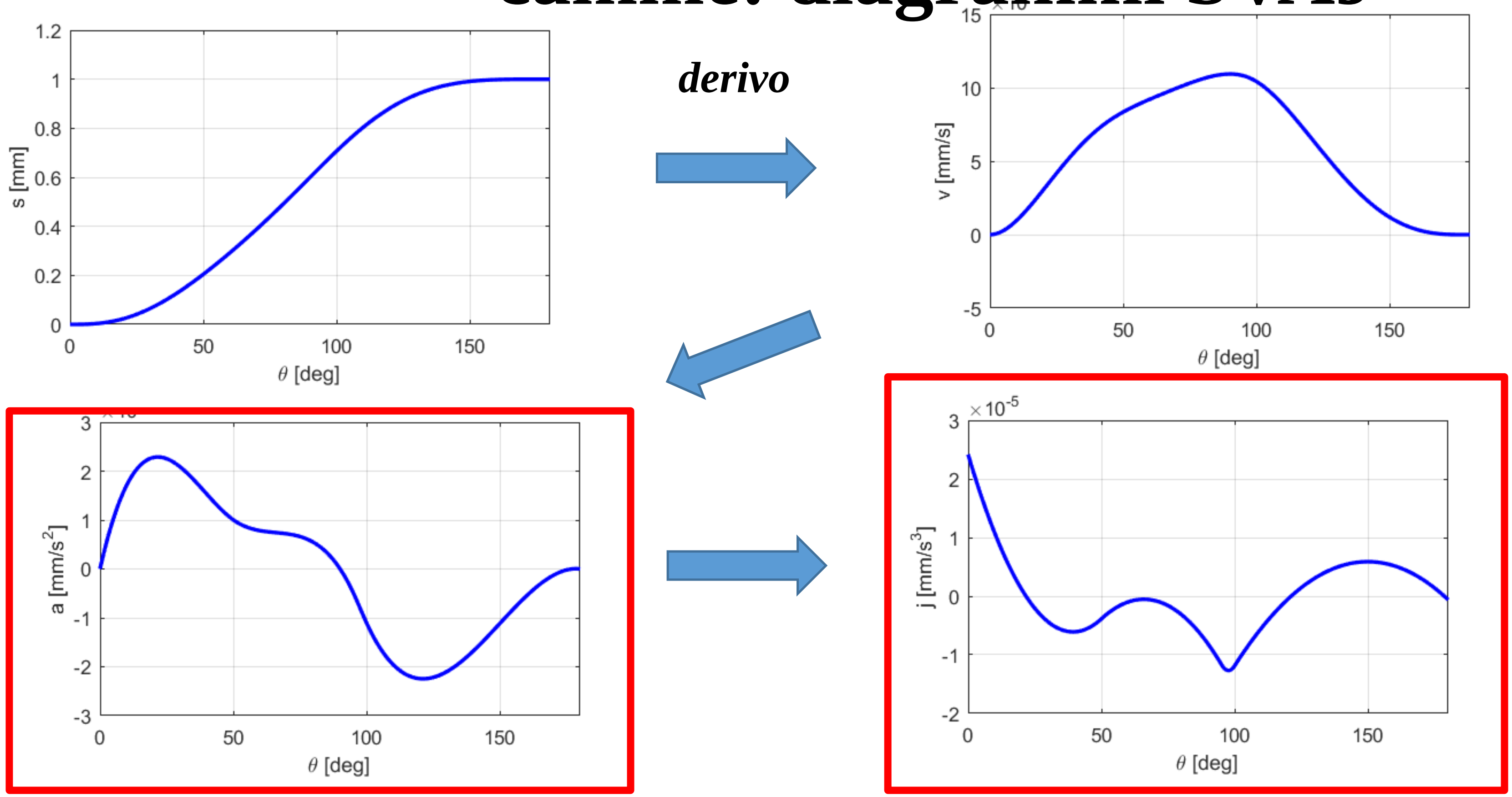


Metodo dell’inviluppo



Università di Roma Tor Vergata

Utilizzo di curve nella sintesi di camme: diagrammi SVAJ



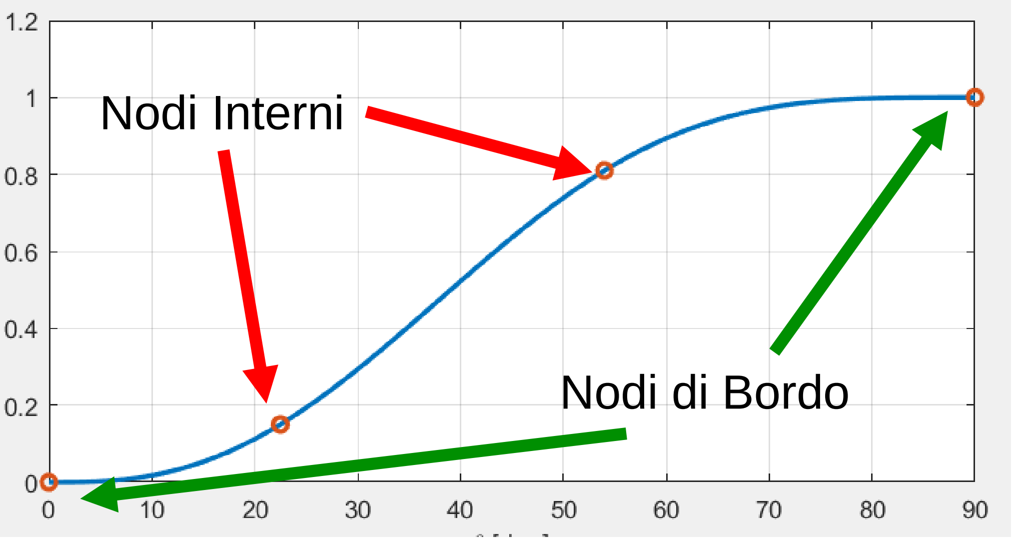
Obiettivo del progettista: Realizzare un diagramma delle alzate compatibile con i dati, che soddisfi dei requisiti cinematici richiesti (accelerazioni massime contenute, velocità nulle in alcuni punti, jerk elevati ridotti)

Spline classica polinomiale

Una spline classica di ordine ***k*** è una curva costruita da diversi polinomi, ognuno di grado ***k-1***, che sono *attaccate* assieme attraverso dei punti definiti ***nodi di interpolazione (interni)***, in modo tale da garantire che la curva sia continua in tutte le sue derivate, fino alla derivate di ordine ***k-2.***

ESEMPIO: Spline di ordine 6

- Grado della spline: 5
- Continuità: spostamento (disp); derivata prima (vel); seconda (acc); terza (jerk); quarta (jounce)

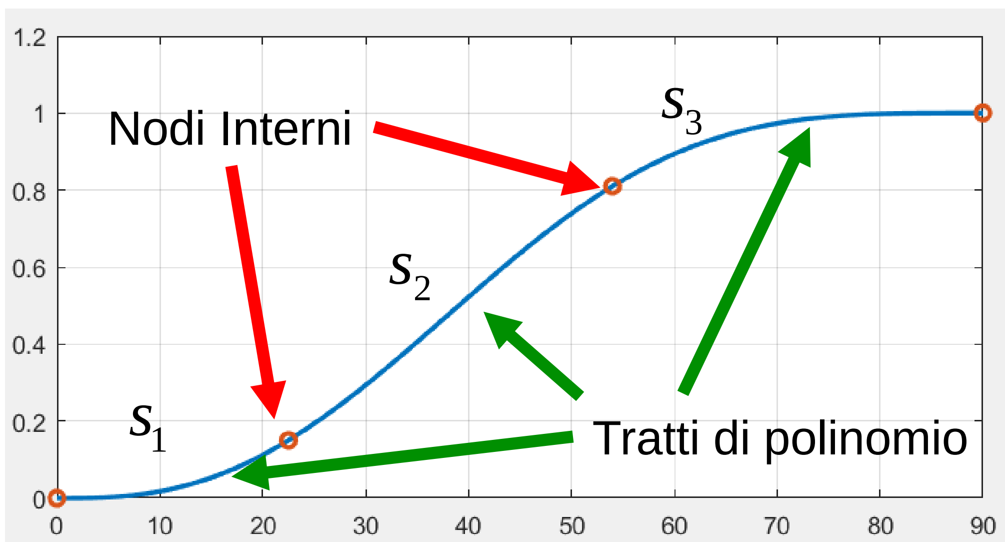


Caso limite N=2: Ho semplicemente i nodi di bordo, quindi una polinomiale semplice.

Spline classica polinomiale

Definito l'intervallo ***[a b]***, si definisce il numero totale di nodi ***N***. La spline polinomiale sarà costituita da ***N-1*** polinomi di ordine ***k.***

$$a = \vartheta_1 \le \vartheta_2 \le \dots \vartheta_N = b \qquad\qquad s(a) = s_1, s_2, \dots s_{N-1} = s(b)$$



$$\boxed{S_j = A_j(\theta-\theta_j)^5 + B_j(\theta-\theta_j)^4 + C_j(\theta-\theta_j)^3 + D_j(\theta-\theta_j)^2 + E_j(\theta-\theta_j) + F_j}$$

Spline classica polinomiale

Confrontando un **polinomio semplice** con una **spline classica polinomiale** si ha che:

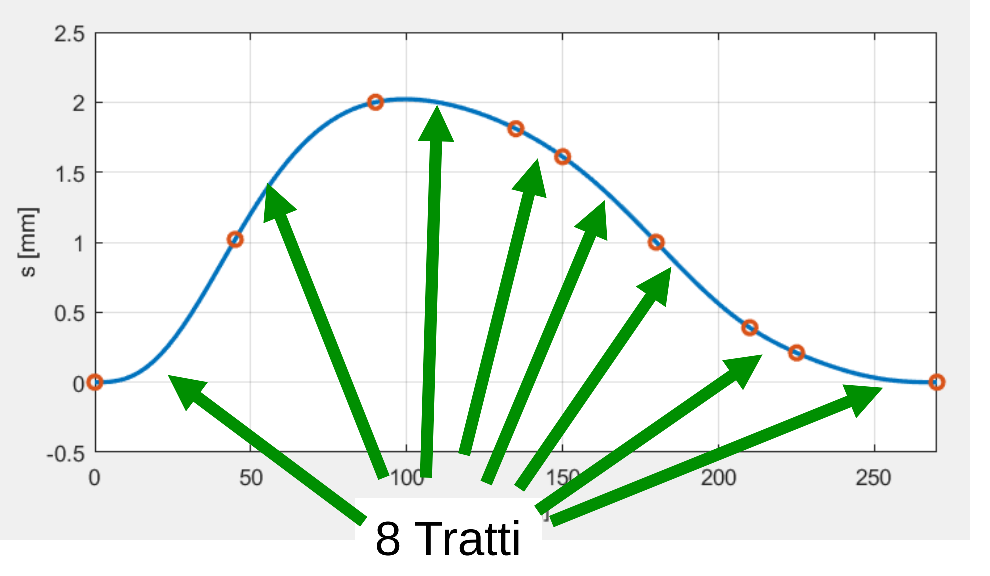
Spline polinomiale di ordine ***k***:
- Numero di nodi: ***N***
- Tratti di polinomio: ***N-1***
- Coefficienti da determinare: ***k(N-1)***

Polinomio ordine ***k***:
- Numero di nodi: ***2***
- Tratti di polinomio: ***1***
- Coefficienti da determinare: ***k***

ESEMPIO: <u>Determinare una curva che passi per 9 punti di interpolazione</u>

Polinomiale: Devo necessariamente usare una curva di ordine 9 (coefficienti)

Spline classica:
Posso interpolare N nodi con un ordine della spline più basso!
Scelgo un ordine (ad esempio 6);
Impongo le condizioni di continità e quelle a contorno;
Avrò 6(9-1) coefficienti da determinare: 48



Università di Roma Tor Vergata

Spline classica polinomiale

Problema inverso:
risaliamo alle condizioni a contorno partendo dal numero delle incognite

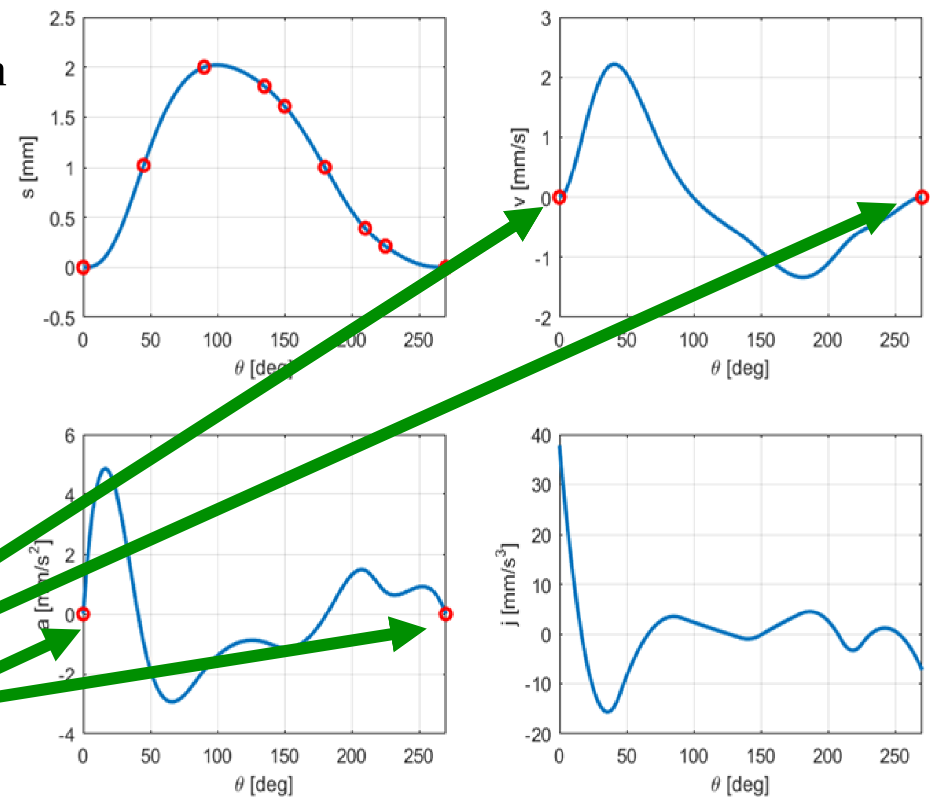
A quali condizioni corrispondono questi coefficienti?

<u>Continuità modi interni:</u>
7 => 7x5 = 35 condizioni sulla continuità (fino alla derivata 4°)
<u>Alzata dei nodi interni interpolanti:</u>
7 condizioni
<u>Alzata dei nodi di bordo interpolanti:</u>
2 condizioni

Restano 4 coefficienti da determinare.

Come si risolve il sistema?

Impongo 4 condizioni a contorno sulle derivate



8 Tratti

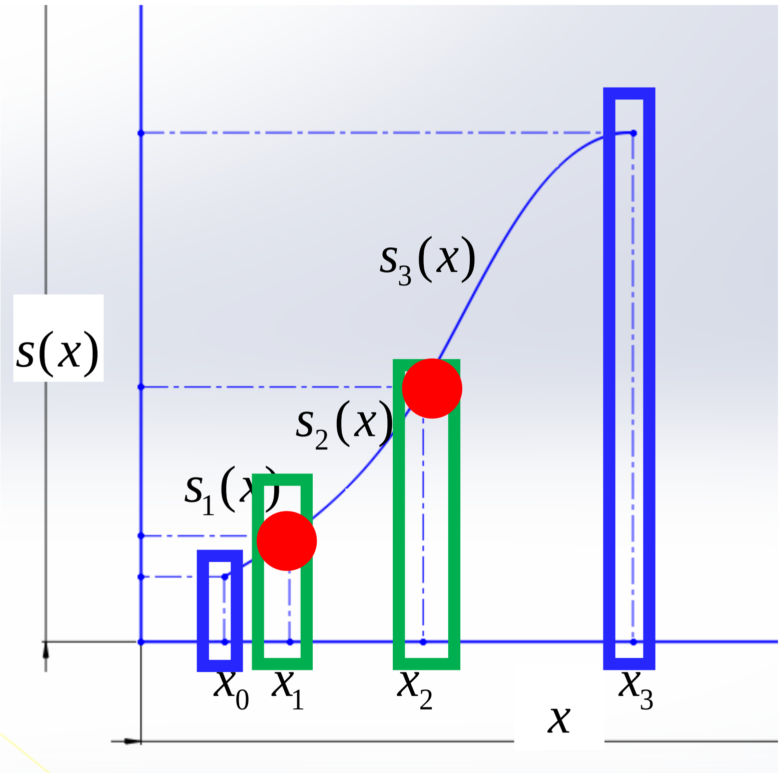
Impostazione del sistema da risolvere

$$S = A_j(\theta-\theta_j)^5 + B_j(\theta-\theta_j)^4 + C_j(\theta-\theta_j)^3 + D_j(\theta-\theta_j)^2 + E_j(\theta-\theta_j) + F_j$$
$$V = 5A_j(\theta-\theta_j)^4 + 4B_j(\theta-\theta_j)^3 + 3C_j(\theta-\theta_j)^2 + 2D_j(\theta-\theta_j) + E_j$$
$$A = 20A_j(\theta-\theta_j)^3 + 12B_j(\theta-\theta_j)^2 + 6C_j(\theta-\theta_j) + 2D_j$$
$$J = 60A_j(\theta-\theta_j)^2 + 24B_j(\theta-\theta_j) + 6C_j$$

$$con \quad j = 1,\ldots,N-1$$

Equazioni SVAJ

Esempio 3 tratti
Ordine 6
Nodi 4



Condizioni di continuità

$$s_i(x_i) = s_{i+1}(x_i)$$
$$s'_i(x_i) = s'_{i+1}(x_i)$$
$$s''_i(x_i) = s''_{i+1}(x_i)$$
$$s'''_i(x_i) = s'''_{i+1}(x_i)$$
$$s''''_i(x_i) = s''''_{i+1}(x_i)$$

Condizioni di alzata iniziale e finale

$$s_1(x_0) = H_0$$
$$s_n(x_n) = H_{fin}$$

Condizioni cinematiche

$$v_1(x_0) = V_1$$
$$v_n(x_n) = V_n$$
$$a_1(x_1) = A_1$$
$$j_n(x_n) = J_n$$

Condizioni di alzata nodi interni

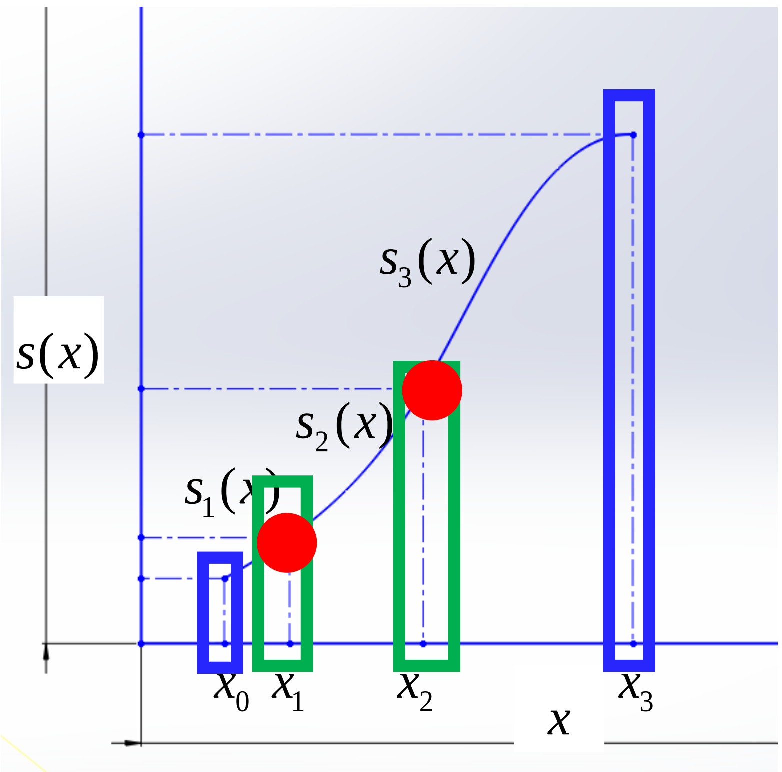
$$s_i(x_i) = H_i$$

Non Visibili nel diagramma

Impostazione del sistema da risolvere

Sistema Lineare da risolvere!

$$\begin{bmatrix} [Continuità] \\ [Alzata\ nodi\ interni] \\ [Alzata\ iniziale\ e\ finale] \\ [Vincoli\ cinematici] \end{bmatrix} \begin{Bmatrix} A_1 \\ B_1 \\ \cdot \\ \cdot \\ \cdot \\ E_{n-1} \\ F_{n-1} \end{Bmatrix} = \begin{Bmatrix} [0] \\ [H_i] \\ [H_{1-fin}] \\ [V; A; J] \end{Bmatrix}$$



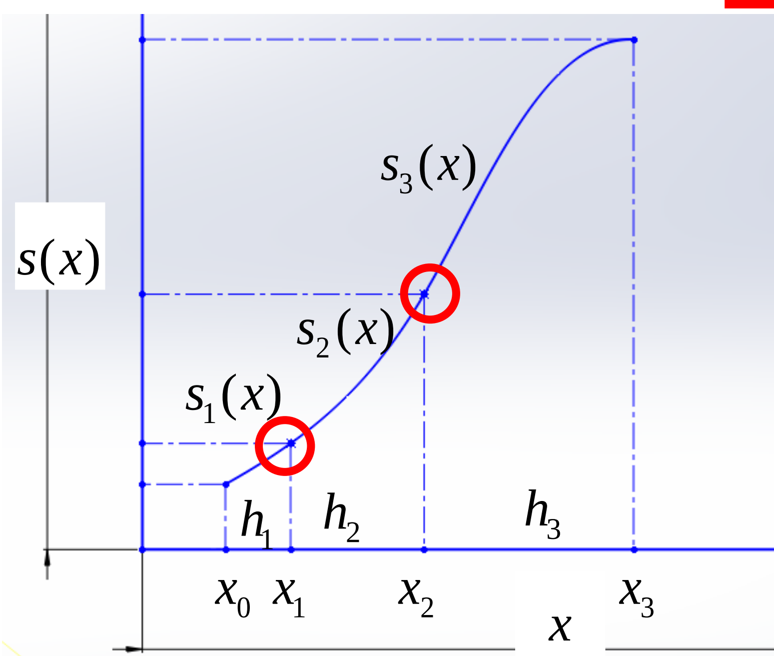
Come si imposta il sistema sopra indicato?

Sviluppiamo l'esempio

Impostazione del sistema da risolvere

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0.2618	0.0685	0.0179	0.0047	0.0012	-1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0.5236	0.2056	0.0718	0.0235	0	-1	0	0	0	0						0
3	0	0	2	1.5708	0.8225	0.3589	0	0	-2	0	0	0						0
4	0	0	0	6	6.2832	4.1123	0	0	0	-6	0	0	0	0	0	0	0	0
5	0	0	0	0	24	31.41...	0	0	0	0	-24	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0.2880	0.0829	0.0239	0.0069	0.0020	-1	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0.5760	0.2488	0.0955	0.0344	0	-1	0	0	0	0
8	0	0	0	0	0	0	0	0	2	1.7279	0.9952	0.4777	0	0	-2	0	0	0
9	0	0	0	0	0	0	0	0	0	6	6.9115	4.9759	0	0	0	-6	0	0
10	0	0	0	0	0	0	0	0	0	0	24	34.55...	0	0	0	0	-24	0

Continuità Cinematica



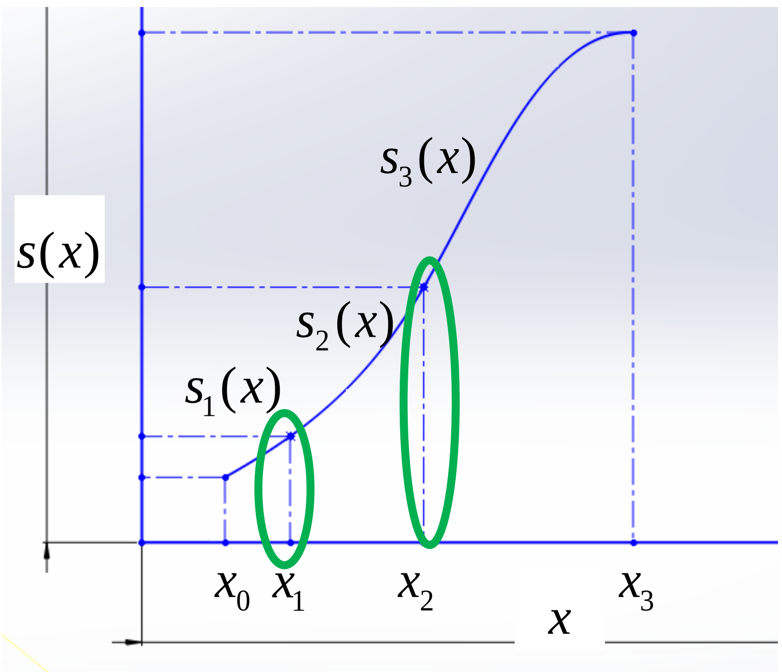
$$S_j = A_j(\theta-\theta_j)^5 + B_j(\theta-\theta_j)^4 + C_j(\theta-\theta_j)^3 + D_j(\theta-\theta_j)^2 + E_j(\theta-\theta_j) + F_j$$

$$j = 1,\dots,n-1 \qquad h_j = x_j - x_{j-1}$$

$$\begin{cases} F_{j+1} = A_j h_j^5 + B_j h_j^4 + C_j h_j^3 + D_j h_j^2 + E_j h_j + F_j \\ E_{j+1} = 5A_j h_j^4 + 4B_j h_j^3 + 3C_j h_j^2 + 2D_j h_j + E_j \\ D_{j+1} = 10A_j h_j^3 + 6B_j h_j^2 + 3C_j h_j + D_j \\ C_{j+1} = 10A_j h_j^2 + 4B_j h_j + C_j \\ B_{j+1} = 5A_j h_j + B_j \end{cases}$$

Impostazione del sistema da risolvere

14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.0821	0.8782	0.6335	0.4285
15	1	0.2618	0.0685	0.0179	0.0047	0.0012	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	1	0.2880	0.0829	0.0239	0.0069	0.0020	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	12.98...	17.56...



Alzate Nodi interni

$$s_1(x_1) = H_1$$

$$s_2(x_2) = H_2$$

$$A_1 h_1^{\,5} + B_1 h_1^{\,4} + C_1 h_1^{\,3} + D_1 h_1^{\,2} + E_1 h_1 + F_1 = H_1$$

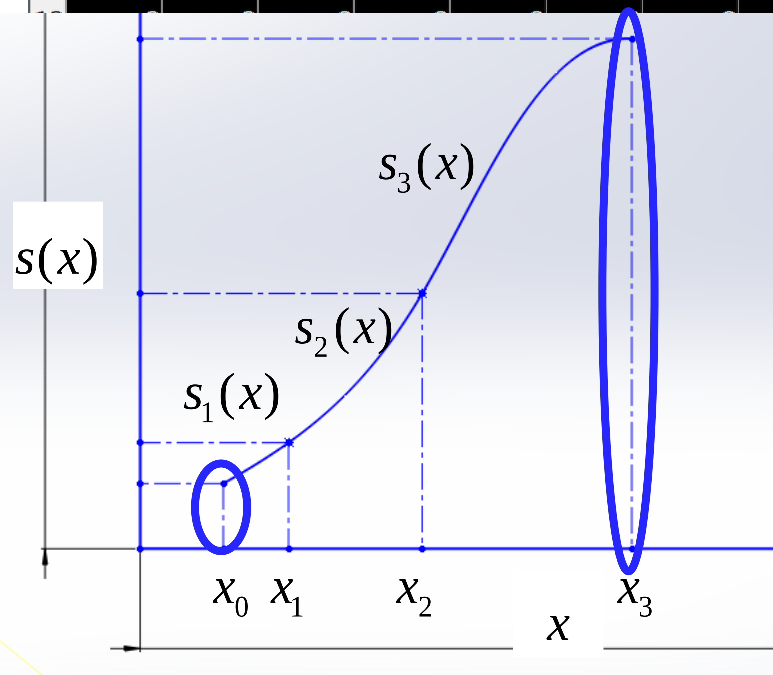
$$A_2 h_2^{\,5} + B_2 h_2^{\,4} + C_2 h_2^{\,3} + D_2 h_2^{\,2} + E_2 h_2 + F_2 = H_2$$

Impostazione del sistema da risolvere

8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	2	1.7279	0.9952	0.4777	0	0	-2	0	0	0
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	6	6.9115	4.9759	0	0	0	-6	0	0
10	0	0	0	0	0	0	0	0	0	0	24	34.55...	0	0	0	0	-24	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.0821	0.8782	0.6335	0.4285
15	1	0.2618	0.0685	0.0179	0.0047	0.0012	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
16	0	0	0	0	0	0	1	0.2880	0.0829	0.0239	0.0069	0.0020	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	12.98...	17.56...
18	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	1	0.5411	0.2927	0.1584	0.0857	0.0464

Alzata Iniziale

Alzata Finale



Alzata Iniziale $s_1(x_0) = H_0$

Alzata Finale $s_3(x_3) = H_3$

$$F_1 = H_0$$

$$A_3 h_3^{\,5} + B_3 h_3^{\,4} + C_3 h_3^{\,3} + D_3 h_3^{\,2} + E_3 h_3 + F_3 = H_3$$

Impostazione del sistema da risolvere

12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.0821	0.8782	0.6335	0.4285
15	1	0.2618	0.0685	0.0179	0.0047	0.0012	0	0	0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0
16	0	0	0	0	0	0	1	0.2880	0.0829	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	12.98...	17.56..
18	0	0	0	0	0	0	0	0	0	0	0	0	1	0.5411	0.2927	0.1584	0.0857	0.0464

Velocità finale

Jerk finale

Velocità iniziale	$s'_1(x_0) = 0$	$E_1 = 0$
Accelerazione iniziale	$s''_1(x_0) = 0$	$D_2 = 0$
Velocità finale	$s'_3(x_3) = 0$	$5A_3h_3^{\,4} + 4B_3h_3^{\,3} + 3C_3h_3^{\,2} + 2D_3h_3 + E_3 = 0$
Jerk finale	$s'''_3(x_3) = 0$	$60A_3h_3^{\,2} + 24B_3h_3 + 6C_3 = 0$

Impostazione del sistema da risolvere

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0.2618	0.0685	0.0179	0.0047	0.0012	-1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0.5236	0.2056	0.0718	0.0235	0	-1	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
3	0	0	2	1.5708	0.8225	0.3589	0	0	-2	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
4	0	0	0	6	6.2832	4.1123	0	0	0	-6	0	0	0	0	0	0	0	0
5	0	0	0	0	24	31.41...	0	0	0	0	-24	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0.2880	0.0829	0.0239	0.0069	0.0020	-1	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0.5760	0.2488	0.0955	0.0344	0	-1	0	0	0	0
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	2	1.7279	0.9952	0.4777	0	0	-2	0	0	0
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	6	6.9115	4.9759	0	0	0	-6	0	0
10	0	0	0	0	0	0	0	0	0	0	24	34.55...	0	0	0	0	-24	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.0821	0.8782	0.6335	0.4285
15	1	0.2618	0.0685	0.0179	0.0047	0.0012	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	1	0.2880	0.0829	0.0239	0.0069	0.0020	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	12.98...	17.56...
18	0	0	0	0	0	0	0	0	0	0	0	0	1	0.5411	0.2927	0.1584	0.0857	0.0464

Continuità Cinematica

Alzata Iniziale

velocità, accelerazione iniziale

Alzata Nodi interni

Alzata Finale

Velocità finale

Jerk finale

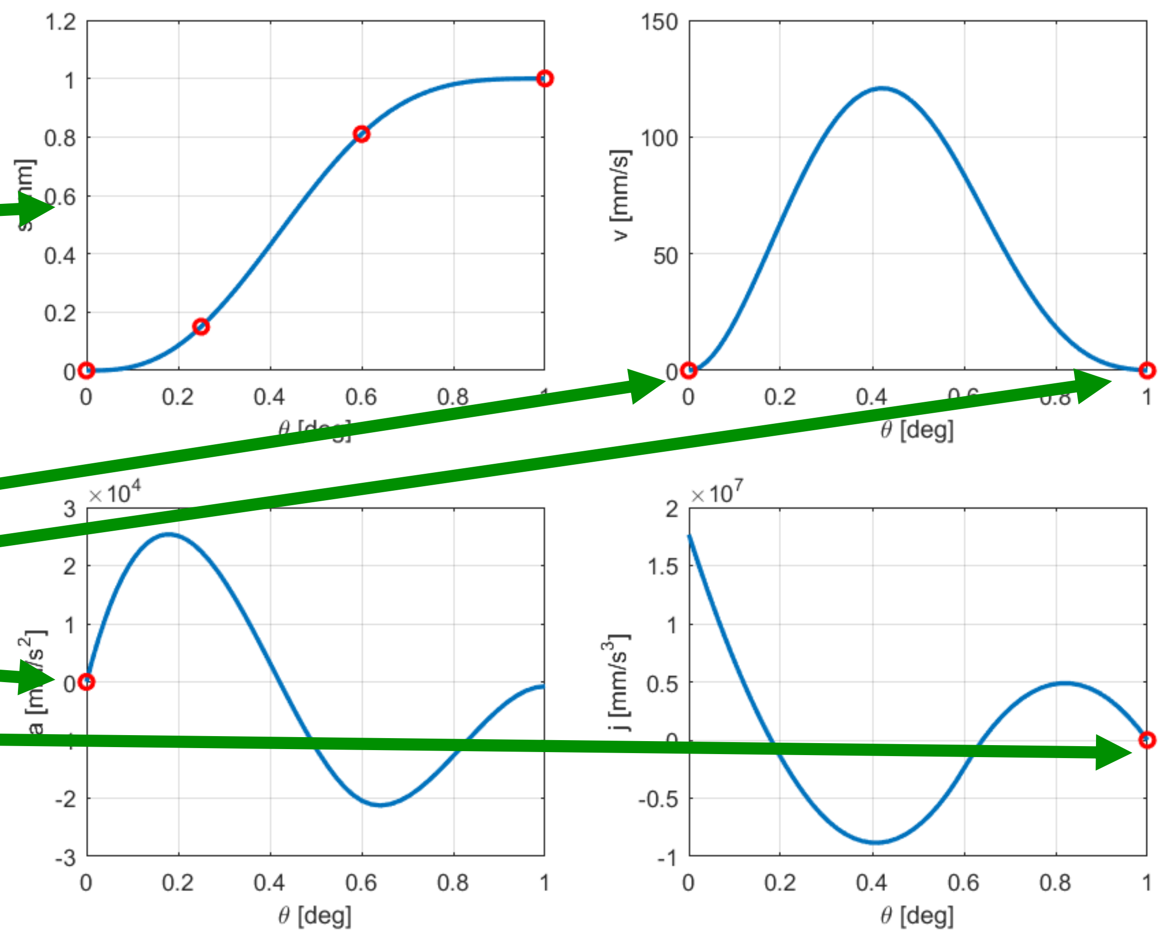
Università di Roma Tor Vergata

Risultato

Prendiamo l'esempio numerico dove:

$$s(x_0) = 0$$
$$s(x_1) = 0.15$$
$$s(x_2) = 0.8$$
$$s(x_3) = 1$$

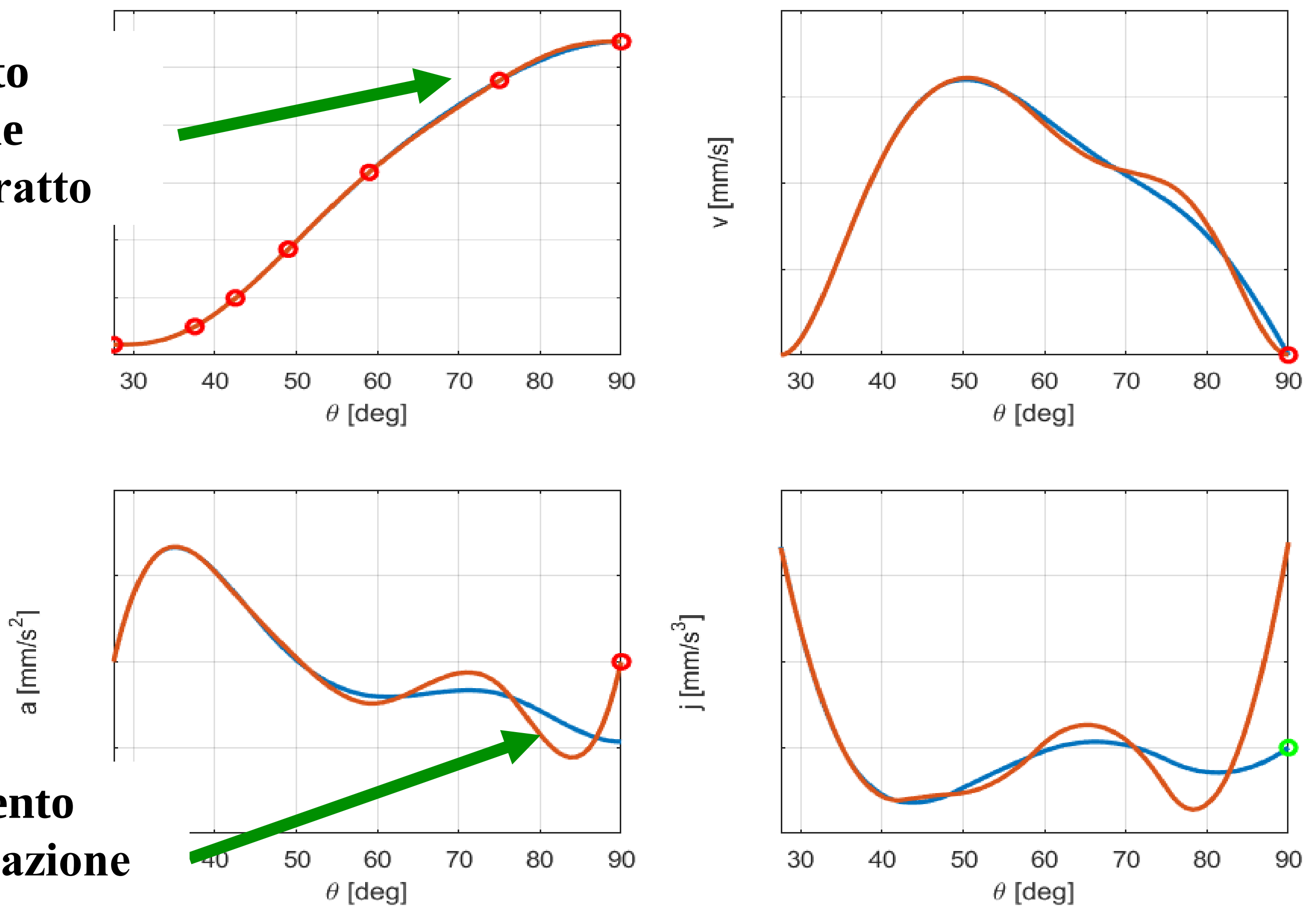
$$s'(x_0) = 0$$
$$s'(x_3) = 0$$
$$s''(x_0) = 0$$
$$s'''(x_3) = 0$$



Nota sulle condizioni a contorno

La scelta delle condizioni a contorno è fondamentale nella sintesi di camme.
Un passaggio dalla condizione **accelerazione finale nulla** a **jerk finale nullo** genera:

Cambiamento impercettibile dell’ultimo tratto



Peggioramento dell’accelerazione negativa

B-Splines

Una B-spline rappresenta una spline costruita utilizzando più curve di Bezier attaccate tra loro.

$$S(x) = \sum_{j=1}^{n} A_j N_{j,k}(x)$$

Funzioni Base

$$S'(x) = \sum_{j=1}^{n} A_j N'_{j,k}(x)$$

Derivata prima delle funzioni base

$$S^{(m)}(x) = \sum_{j=1}^{n} A_j N^{(m)}{}_{j,k}(x)$$

Derivata m-esima delle funzioni base

Coefficienti da determinare

Cosa cambia rispetto alle spline classiche?

Metodo di implementazione
Il numero dei vincoli cinematici da imporre, non dipende dal’ordine della spline

Approccio SISTEMATICO

B-Splines: Modalità operativa

$$S(x)=\sum_{j=1}^{n} A_j N_{j,k}(x)$$

Fase 1

1) Si definisce il numero di vincoli cinematici che si vuole imporre ***n***
2) Si definisce l'ordine della B-spline che si vuole utilizzare ***k***
3) Si genera il vettore dei nodi ***T*** nell'intervallo di esistenza della spline $[x_{\min}, x_{\max}]$

   Tale vettore sarà del tipo $[T_1, T_2, \ldots, T_{k+n}]$

Fase 2

4) Si imposta il sistema di equazioni da risolvere per identificare i coefficienti Aj

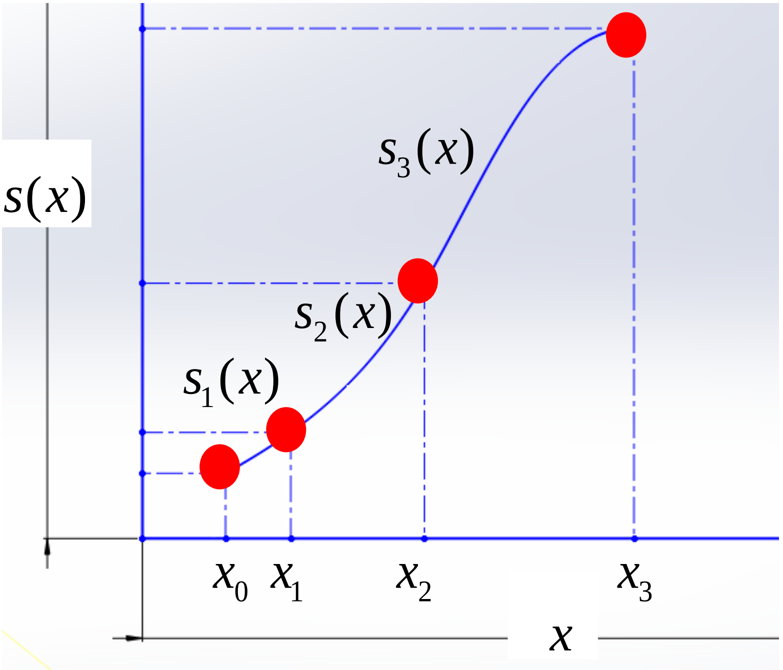
Fase 3

5) Si identificano le funzioni base con un algoritmo ricorsivo (de Boor e Cox)
6) Si valutano le derivate delle funzioni base con un algoritmo simile

Fase 4

7) Si risolve il sistema e si determinano le spline di spostamento, velocità, accelerazione e jerk.

B-Splines: Fase 1



$$s(x_0) = H_0$$
$$s(x_1) = H_1$$
$$s(x_2) = H_2$$
$$s(x_3) = H_3$$
$$s'(x_0) = 0$$
$$s''(x_0) = 0$$
$$s'(x_3) = 0$$
$$s'''(x_3) = 0$$

Numero totale di vincoli cinematici ***n***:

- 4 spostamenti (alzate)
- 4 condizioni cinematiche sulle derivate

Scegliamo l'ordine della spline ***k***:
Per avere anche in questo caso continuità della derivata del jerk, scegliamo 6

Il vettore dei nodi avrà dimensione: ***k+n=14***

Come si compone il vettore dei nodi?

Definito il campo di esistenza della spline $[x_{min}, x_{max}]$ → $[0,1]$
Si sceglie un vettore con elementi non decrescenti

$$T = [\underbrace{0,0,0,0,0,0,}_{\text{Dimensione } k}\ \underbrace{0.3, 0.55,}_{\text{Dimensione } k-n}\ \underbrace{1,1,1,1,1,1}_{\text{Dimensione } k}];$$

n > k

B-Splines: Fase 2

Concettualmente più complesso;

Approccio sistematico che ben si presta all'implementazione

$$S(x)=\sum_{j=1}^{n}A_jN_{j,k}(x)$$

$$S'(x)=\sum_{j=1}^{n}A_jN'_{j,k}(x)$$

$$S^{(m)}(x)=\sum_{j=1}^{n}A_jN^{(m)}_{j,k}(x)$$

$$\begin{bmatrix} [Spostamento] \\ [Velocità] \\ [Accelerazione] \\ [Jerk] \\ \text{Al più ordine } \boldsymbol{k\text{-}2} \end{bmatrix} \begin{Bmatrix} A_1 \\ . \\ . \\ . \\ A_n \end{Bmatrix} = \begin{Bmatrix} BC_{spost} \\ BC_{vel} \\ BC_{acc} \\ BC_{jerk} \\ . \end{Bmatrix}$$

Esempio:

$n=8$

$k=6$

$s(x_0)=H_0$

$s(x_1)=H_1$

$s(x_2)=H_2$

$s(x_3)=H_3$

$s'(x_0)=0$

$s'(x_3)=0$

$s''(x_0)=0$

$s'''(x_3)=0$

$$\begin{bmatrix} \begin{bmatrix} N_{1,6}(x_0) & N_{2,6}(x_0) & \dots & N_{8,6}(x_0) \\ N_{1,6}(x_1) & N_{2,6}(x_1) & \dots & N_{2,6}(x_1) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ N'_{1,6}(x_0) \quad N'_{2,6}(x_0) \quad \dots \quad N'_{8,6}(x_0) \\ N'_{1,6}(x_3) \quad N'_{2,6}(x_3) \quad \dots \quad N'_{8,6}(x_3) \\ N''_{1,6}(x_0) \quad N''_{2,6}(x_0) \quad \dots \quad N''_{8,6}(x_0) \\ N'''_{1,6}(x_3) \quad N'''_{2,6}(x_3) \quad \dots \quad N'''_{8,6}(x_3) \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

B-Splines: Fase 3

$$S(x) = \sum_{j=1}^{n} A_j N_{j,k}(x)$$

Come si identificano le funzioni B-Splines?

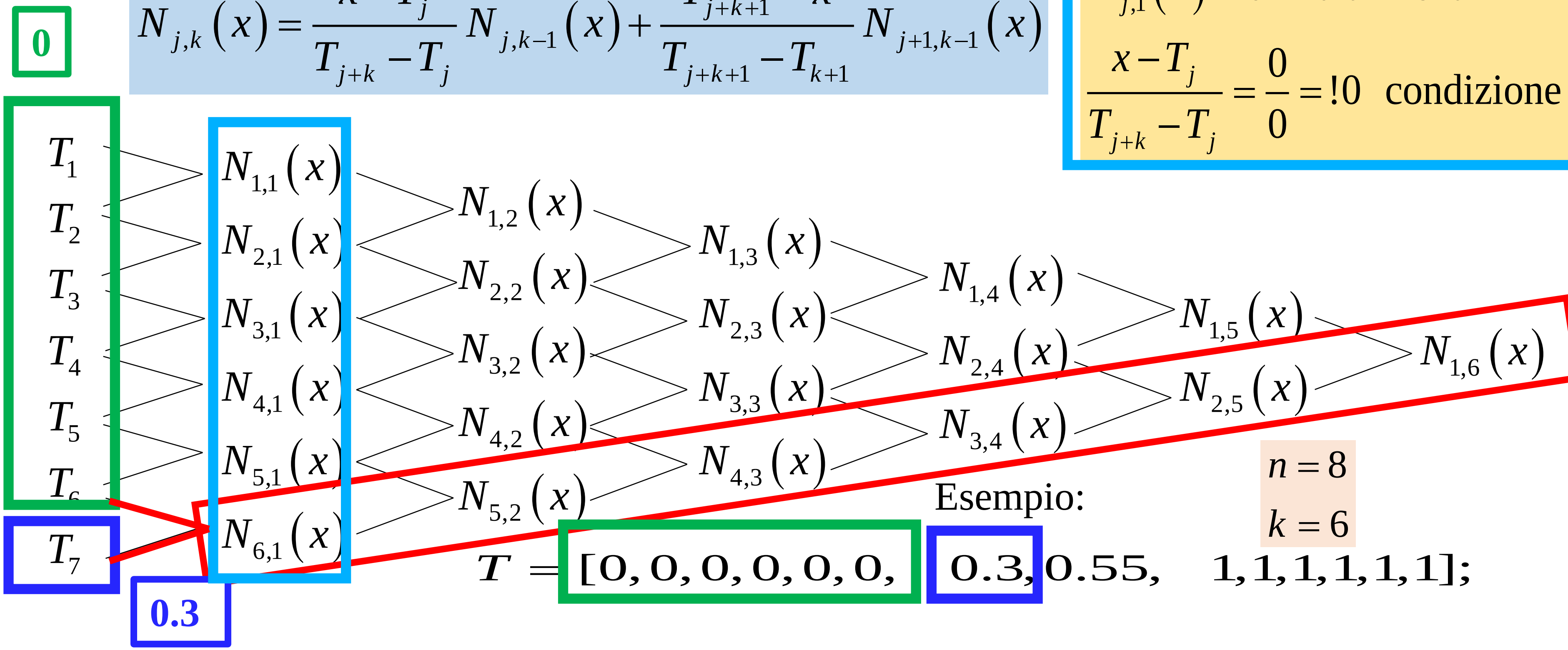
Si utilizza l'algoritmo ricorsivo di Cox-de Boor

$$N_{j,k}(x) = \frac{x - T_j}{T_{j+k} - T_j} N_{j,k-1}(x) + \frac{T_{j+k+1} - x}{T_{j+k+1} - T_{k+1}} N_{j+1,k-1}(x)$$

$N_{j,1}(x) = 1 \quad T_j \leq x < T_{j+1}$

$N_{j,1}(x) = 0 \quad$ altrimenti

$\frac{x - T_j}{T_{j+k} - T_j} = \frac{0}{0} = !0$ condizione



Esempio:

$n = 8$

$k = 6$

$T = [0, 0, 0, 0, 0, 0, \quad 0.3, 0.55, \quad 1, 1, 1, 1, 1, 1];$

B-Splines: Fase 3

$\vdots$

0 $\boxed{0.3}$

$\boxed{T_7}$

$\boxed{T_8}$

1 $\boxed{0.55}$

$\vdots$

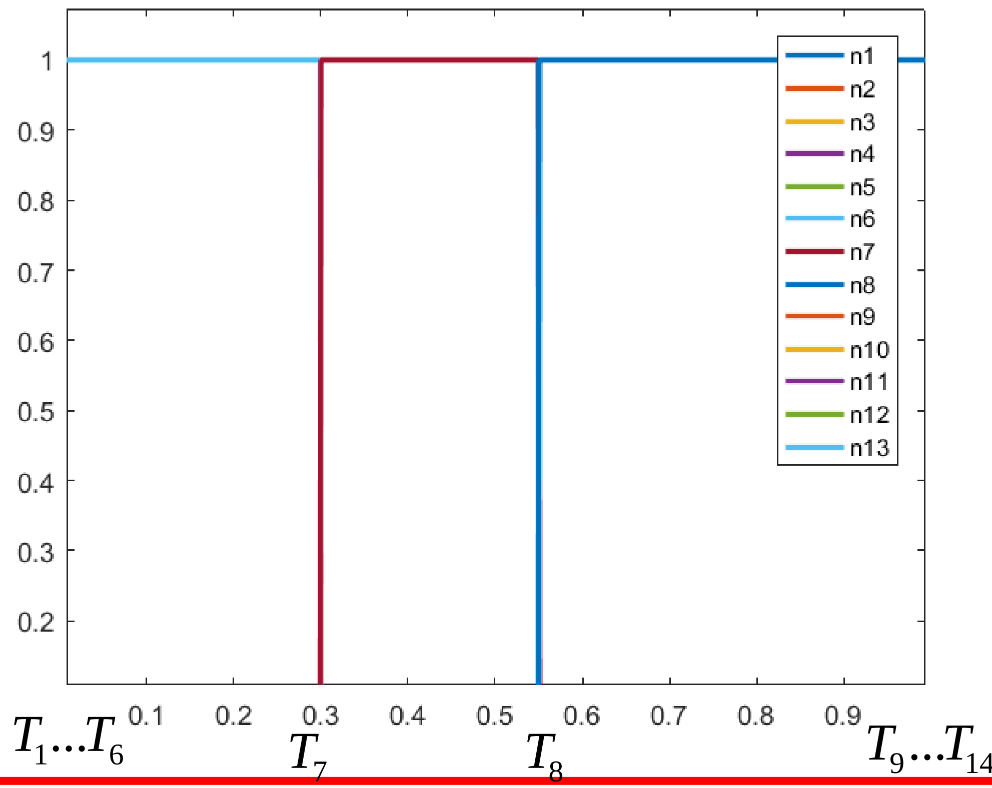
$\boxed{0}$

$\vdots$

$$\boxed{\begin{matrix} N_{6,1}(x) \\ N_{7,1}(x) \\ N_{8,1}(x) \end{matrix}}$$

$\vdots$

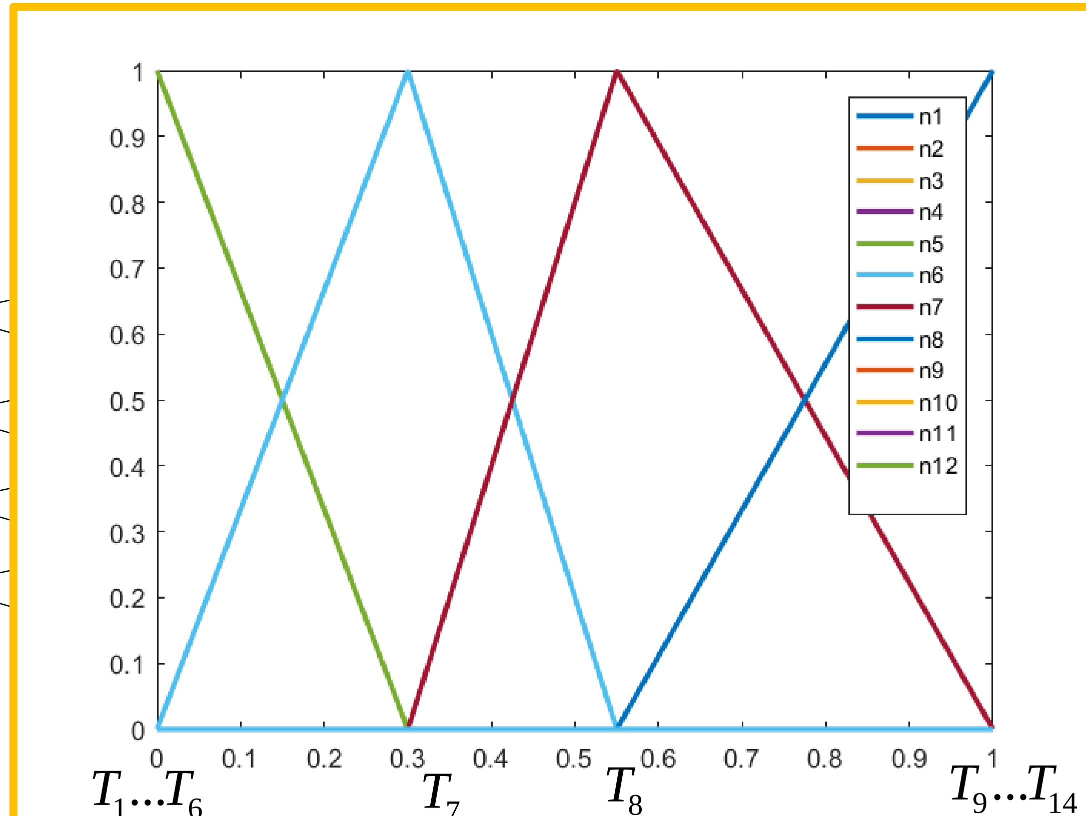
$\boxed{0}$



$$\frac{x - T_j}{T_{j+k} - T_j} = \frac{0}{0} = !0 \quad \text{condizione}$$

B-Splines: Fase 3

$$\begin{array}{cccc}
\vdots & \vdots & & \\
0 & 0 & \boxed{0} & \\
\boxed{0.3} & N_{6,1}(x) & N_{5,2}(x) & \\
\boxed{T_7} & & N_{6,2}(x) & \\
 & N_{7,1}(x) & & \\
\boxed{T_8} & & N_{7,2}(x) & \\
 & N_{8,1}(x) & N_{8,2}(x) & \\
1 \ \boxed{0.55} & 0 & \boxed{0} & \\
\vdots & \vdots & &
\end{array}$$

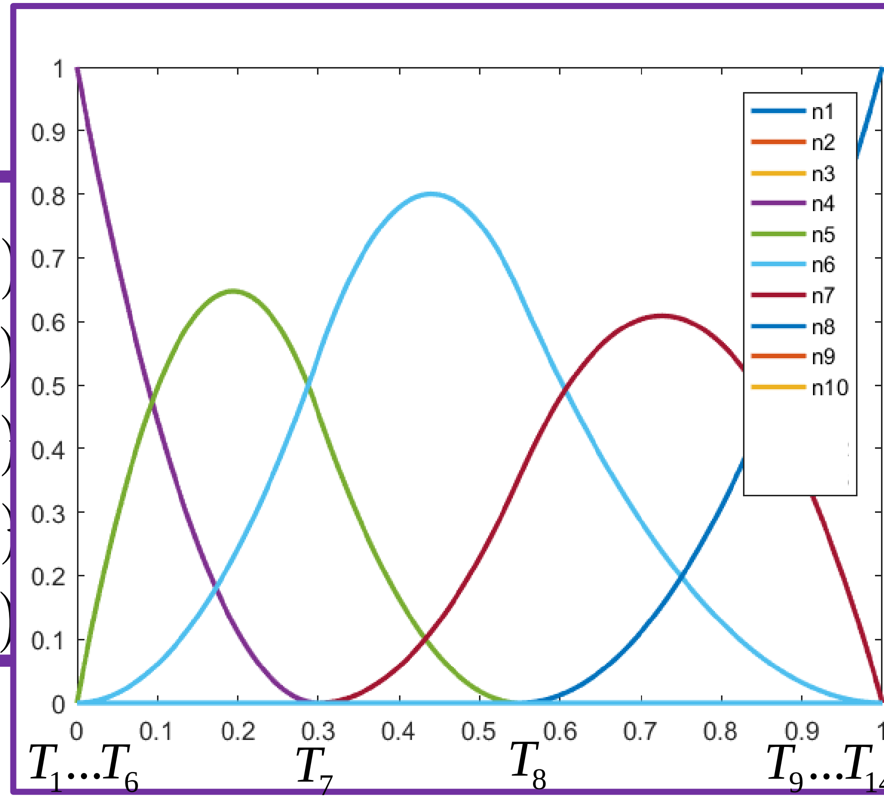


Andamento lineare perché il termine evidenziato aumenta in grado

$$N_{j,k}(x) = \frac{x - T_j}{T_{j+k} - T_j} N_{j,k-1}(x) + \frac{T_{j+k+1} - x}{T_{j+k+1} - T_{k+1}} N_{j+1,k-1}(x)$$

B-Splines: Fase 3

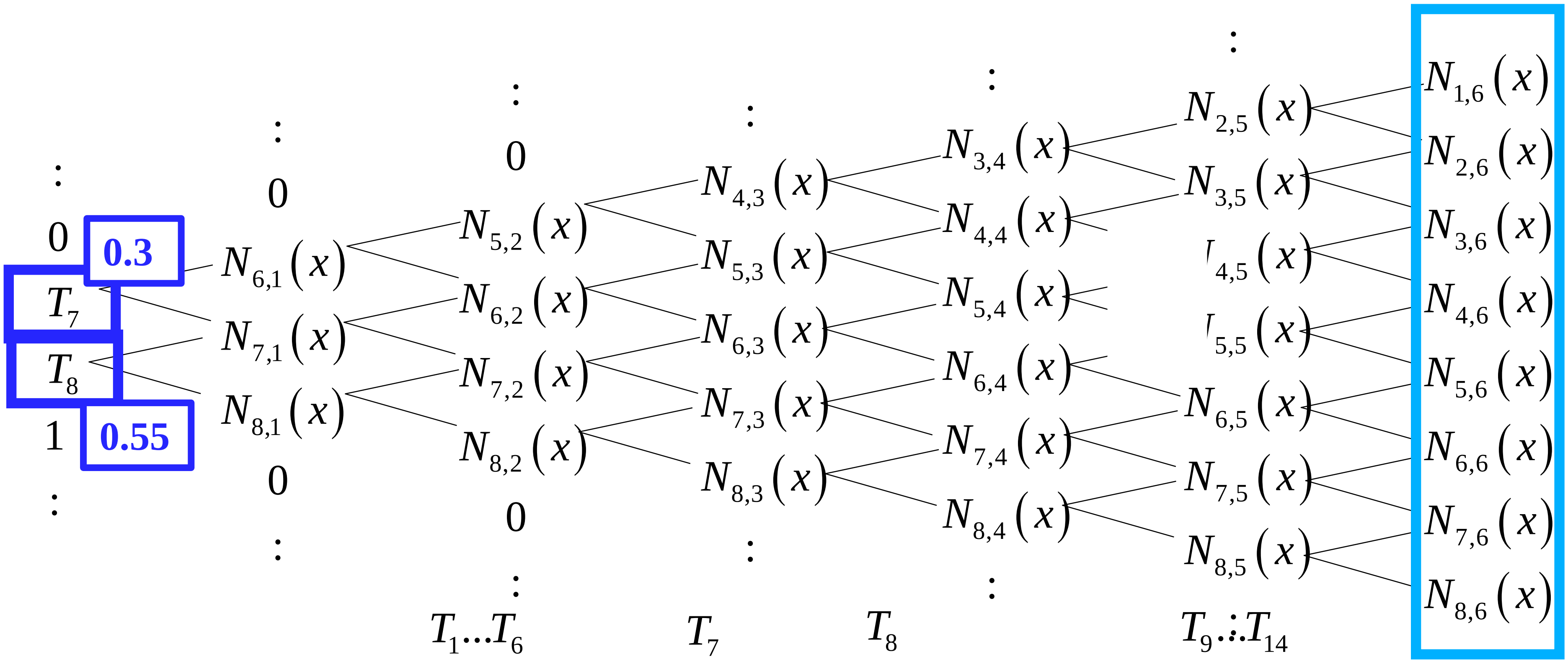
$$
\begin{array}{cccc}
\vdots & & & \\
0 \quad \boxed{0.3} & \vdots & \vdots & \boxed{0} \\
\boxed{T_7} & 0 & 0 & N_{4,3}(x) \\
\boxed{T_8} & N_{6,1}(x) & N_{5,2}(x) & N_{5,3}(x) \\
1 \quad \boxed{0.55} & N_{7,1}(x) & N_{6,2}(x) & N_{6,3}(x) \\
\vdots & N_{8,1}(x) & N_{7,2}(x) & N_{7,3}(x) \\
 & 0 & N_{8,2}(x) & N_{8,3}(x) \\
 & \vdots & 0 & \boxed{0} \\
 & & \vdots &
\end{array}
$$



Andamento di secondo grado

$$
N_{j,k}(x) = \frac{x - T_j}{T_{j+k} - T_j} N_{j,k-1}(x) + \frac{T_{j+k+1} - x}{T_{j+k+1} - T_{k+1}} N_{j+1,k-1}(x)
$$

B-Splines: Fase 3



Andamento di quinto grado

$$N_{j,k}(x) = \frac{x - T_j}{T_{j+k} - T_j} N_{j,k-1}(x) + \frac{T_{j+k+1} - x}{T_{j+k+1} - T_{k+1}} N_{j+1,k-1}(x)$$

B-Splines: Fase 3

$\vdots$

0 $\boxed{0.3}$

$\boxed{T_7}$

$\boxed{T_8}$

1 $\boxed{0.55}$

$\vdots$

$\vdots$

0

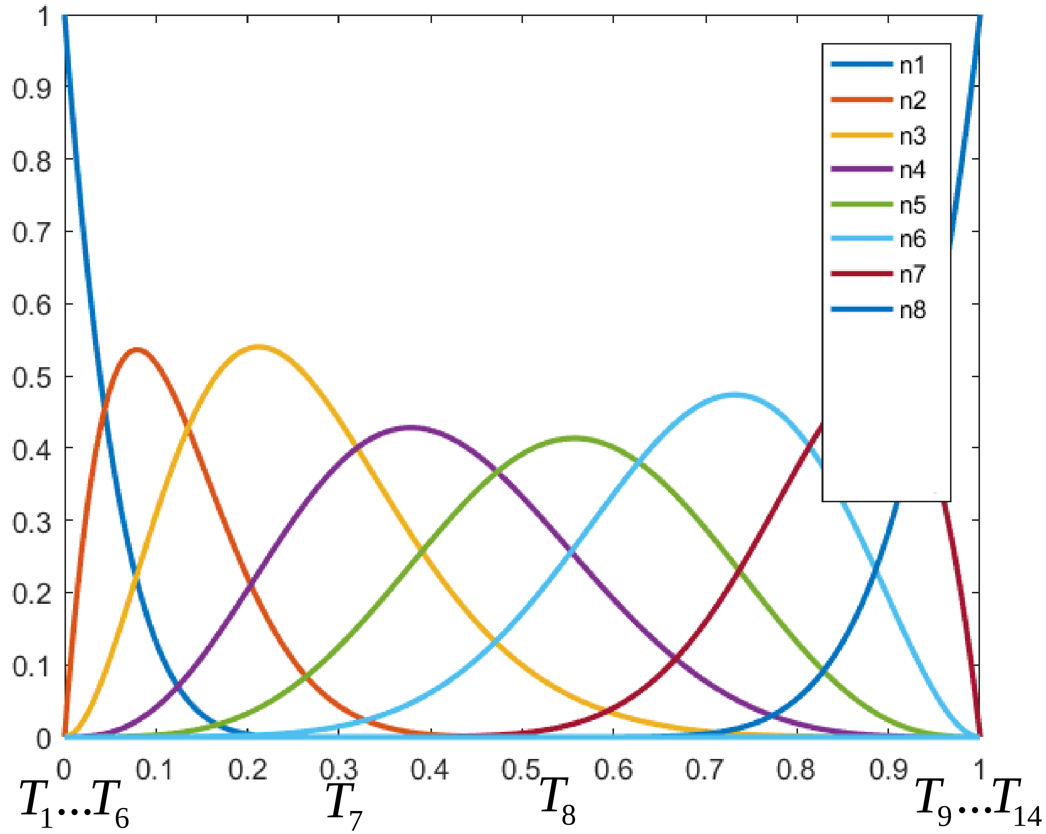
$N_{6,1}(x)$

$N_{7,1}(x)$

$N_{8,1}(x)$

0

$\vdots$



$N_{1,6}(x)$
$N_{2,6}(x)$
$N_{3,6}(x)$
$N_{4,6}(x)$
$N_{5,6}(x)$
$N_{6,6}(x)$
$N_{7,6}(x)$
$N_{8,6}(x)$

Andamento di quinto grado

$$N_{j,k}(x) = \frac{x - T_j}{T_{j+k} - T_j} N_{j,k-1}(x) + \frac{T_{j+k+1} - x}{T_{j+k+1} - T_{k+1}} N_{j+1,k-1}(x)$$

B-Splines: Fase 3

Calcolo delle derivate delle funzioni di base:

$$S(x)=\sum_{j=1}^{n} A\,\boxed{N_{j,k}(x)}$$

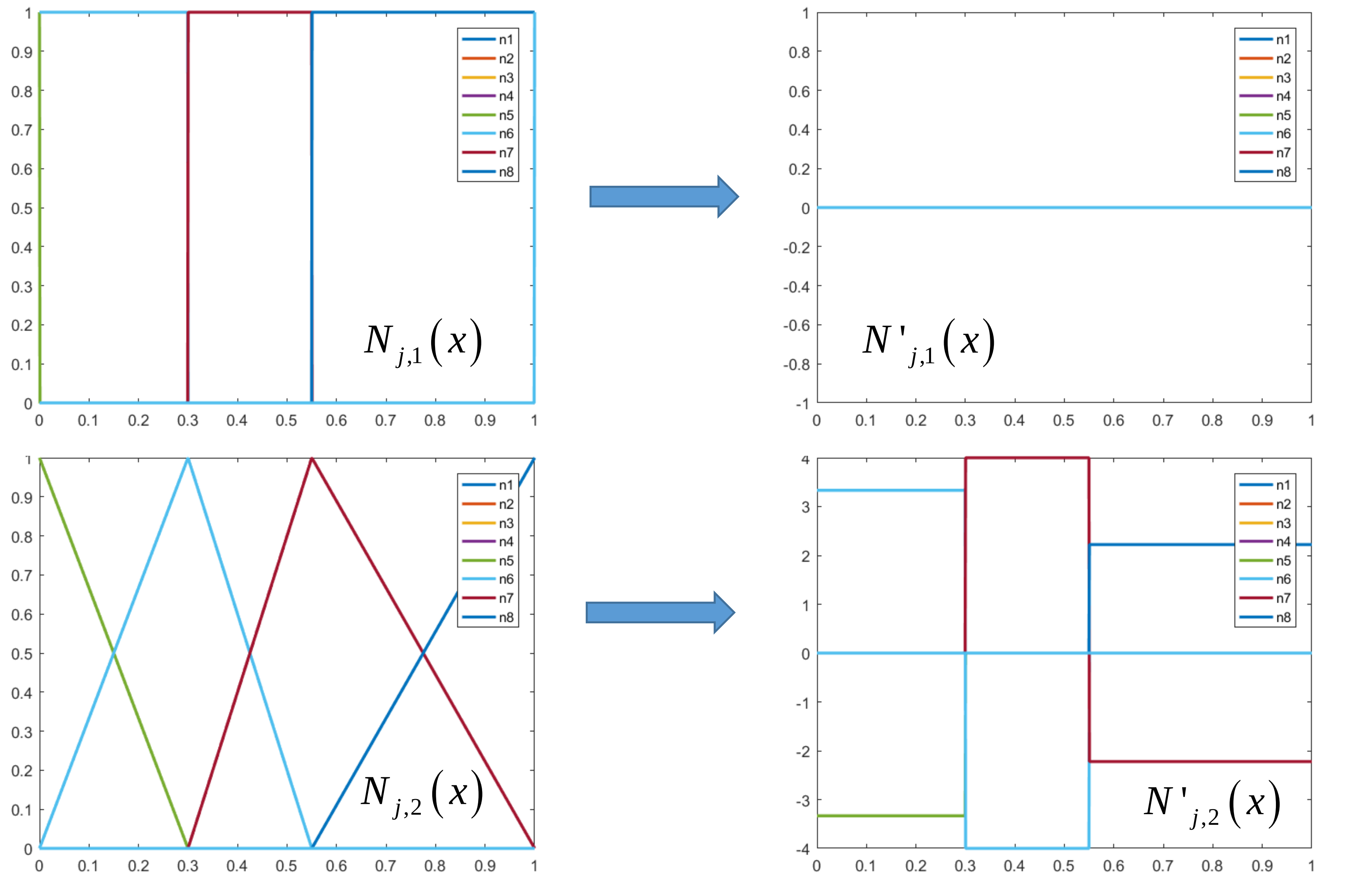
$$S'(x)=\sum_{j=1}^{n} A\,\boxed{N'_{j,k}(x)}$$

$$S^{(m)}(x)=\sum_{j=1}^{n} A\,\boxed{N^{(m)}{}_{j,k}(x)}$$

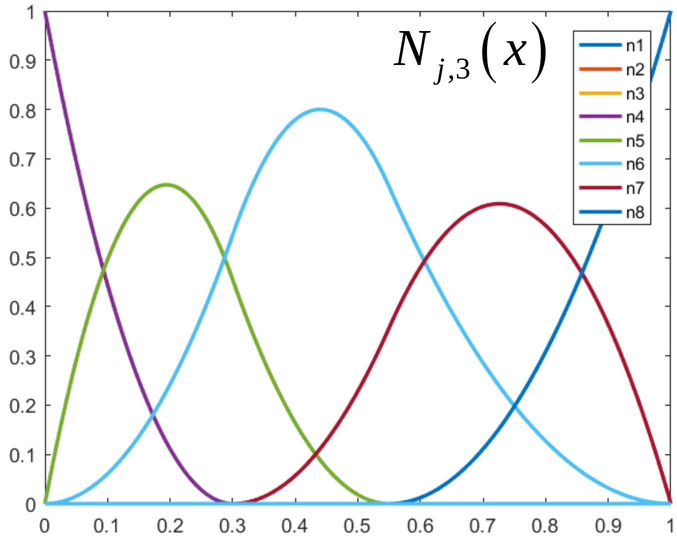
$$N_{j,k}(x)=\frac{x-T_j}{T_{j+k}-T_j}N_{j,k-1}(x)+\frac{T_{j+k+1}-x}{T_{j+k+1}-T_{k+1}}N_{j+1,k-1}(x)$$

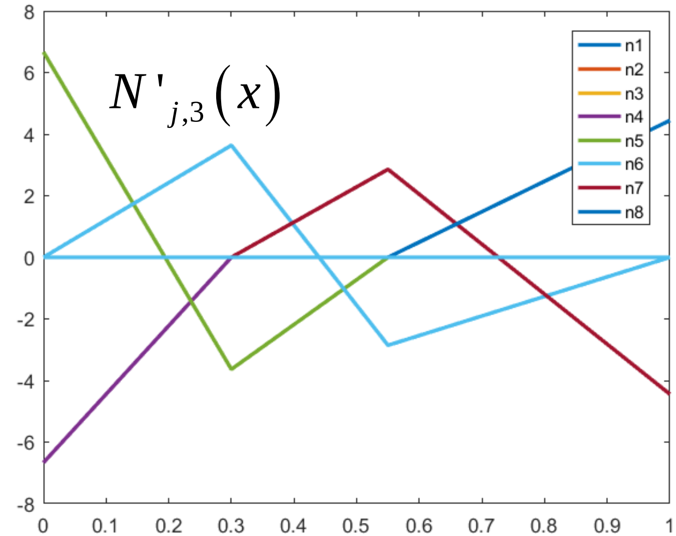
$$N^{(m)}{}_{j,k}(x)=(k-1)\left[\frac{N^{(m-1)}{}_{j,k-1}(x)}{T_{j+k}-T_j}+\frac{N^{(m-1)}{}_{j+1,k-1}(x)}{T_{j+k+1}-T_{k+1}}\right]$$

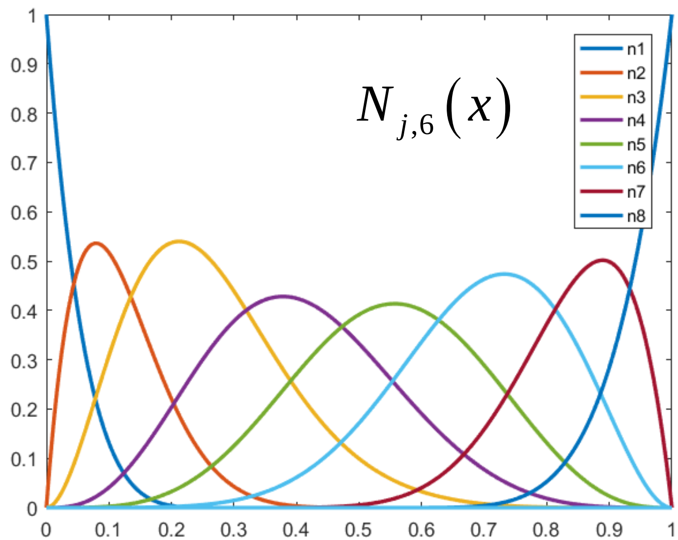
B-Splines: Fase 3

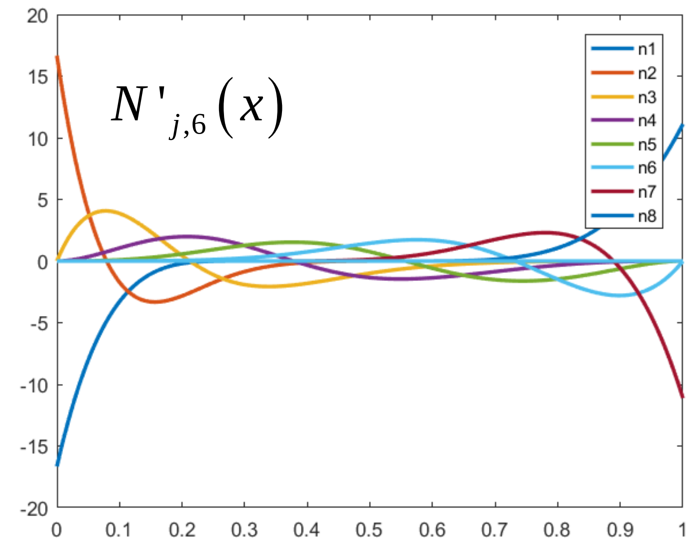


B-Splines: Fase 3









B-Splines: Fase 4

Si risolve il sistema, si identificano i coefficienti

$$
\begin{bmatrix}
\begin{bmatrix}
N_{1,6}(x_0) & N_{2,6}(x_0) & \dots & N_{8,6}(x_0) \\
N_{1,6}(x_1) & N_{2,6}(x_1) & \dots & N_{2,6}(x_1) \\
\vdots & \vdots & \vdots & \vdots
\end{bmatrix} \\
\begin{bmatrix}
N'_{1,6}(x_0) & N'_{2,6}(x_0) & \dots & N'_{8,6}(x_0) \\
N'_{1,6}(x_3) & N'_{2,6}(x_3) & \dots & N'_{8,6}(x_3)
\end{bmatrix} \\
\begin{bmatrix}
N''_{1,6}(x_0) & N''_{2,6}(x_0) & \dots & N''_{8,6}(x_0)
\end{bmatrix} \\
N'''_{1,6}(x_3) \quad N'''_{2,6}(x_3) \quad \dots \quad N'''_{8,6}(x_3)
\end{bmatrix}
\begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \end{Bmatrix}
=
\begin{Bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}
$$

Si risolve il sistema Lineare

$$\{A\}$$

Possiamo calcolare le splines di spostamento, velocità, accelerazione e jerk!!!

$$S(x) = \sum_{j=1}^{n} A_j N_{j,k}(x)$$

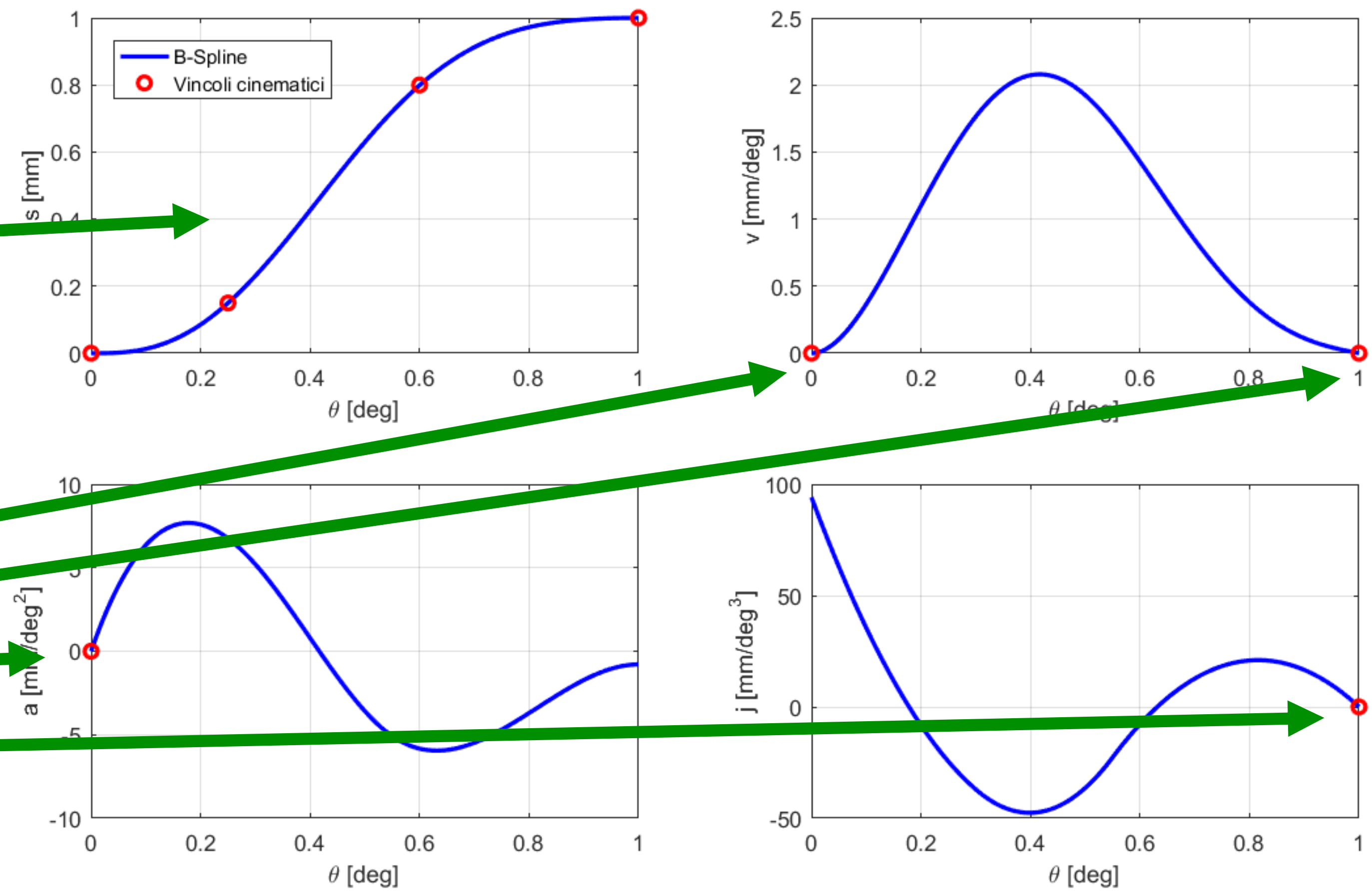
$$S'(x) = \sum_{j=1}^{n} A_j N'_{j,k}(x)$$

$$S^{(m)}(x) = \sum_{j=1}^{n} A_j N^{(m)}_{j,k}(x)$$

Risultato numerico

$$s(x_0) = 0$$
$$s(x_1) = 0.15$$
$$s(x_2) = 0.8$$
$$s(x_3) = 1$$

$$s'(x_0) = 0$$
$$s'(x_3) = 0$$
$$s''(x_0) = 0$$
$$s'''(x_3) = 0$$



NURBS

Una NURBS (Non Uniform Rational B-spline) rappresenta una B-spline con vettore dei nodi non uniforme e caratterizzate dalla presenza di pesi.

$$S(x) = \sum_{j=1}^{n} A_j N_{j,k}(x)$$

$$S(x) = \sum_{j=1}^{n} R_{j,k}(x) A_j \qquad T_1 \leq x \leq T_{k+n}$$

$$S'(x) = \sum_{j=1}^{n} R'_{j,k}(x) A_j$$

$$S^{(m)}(x) = \sum_{j=1}^{n} R^{(m)}{}_{j,k}(x) A_j$$

Le derivate delle nuove funzioni di base non sono immediate.

$$R_{j,k}(x) = \frac{N_{j,k}(x) W_j}{\sum_{j=1}^{n} N_{j,k}(x) W_j}$$

Pesi

Stesse funzioni base delle B-Spline! E SONO DI ORDINE ***k***

Sui pesi:

Il vettore dei pesi sarà un vettore avente dimensione pari al numero di vincoli cinematici che si vuole imporre!

Università di Roma Tor Vergata

NURBS: Modalità operativa

$$S(x) = \sum_{j=1}^{n} R_{j,k}(x) A_j$$

Fase 1

1) Si definisce il numero di vincoli cinematici che si vuole imporre ***n***
2) Si definisce l'ordine della B-spline che si vuole utilizzare ***k***
3) Si genera il vettore dei nodi ***T*** nell'intervallo di esistenza della spline $[x_{\min}, x_{\max}]$
   Tale vettore sarà del tipo $[T_1, T_2, ..., T_{k+n}]$
4) **Si sceglie il vettore dei pesi in base ai vincoli scelti** $[W_1, W_2, ..., W_n]$

Fase 2

5) Si imposta il sistema di equazioni da risolvere per identificare i coefficienti Aj

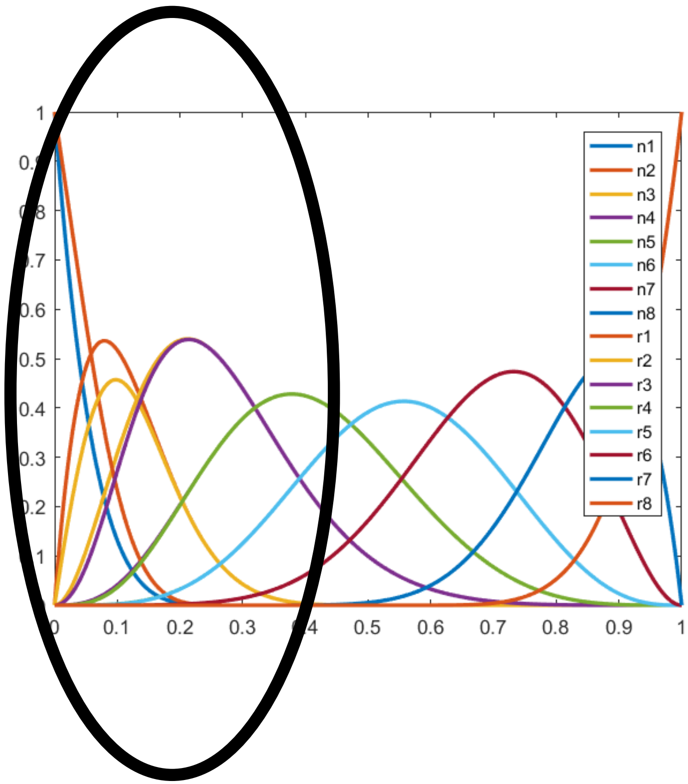
Fase 3

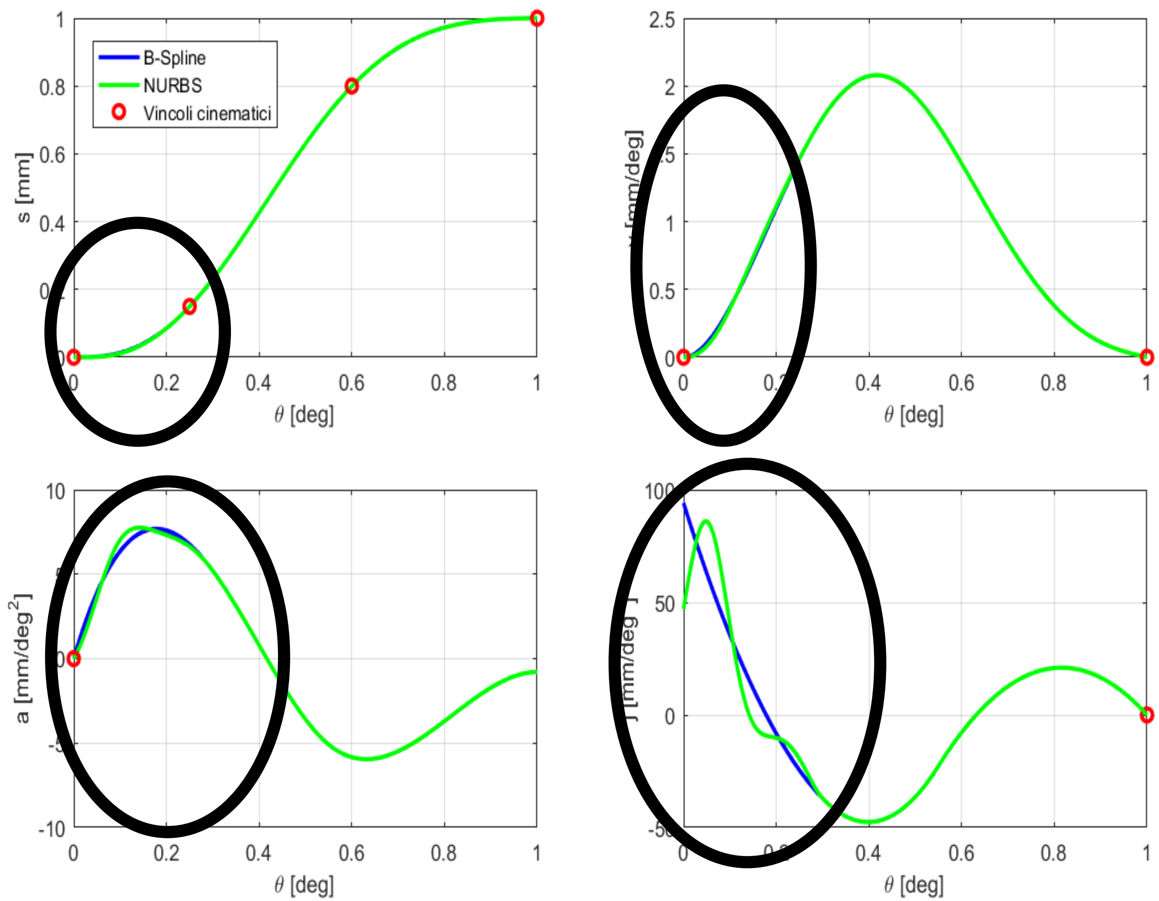
6) Si identificano le funzioni base con un algoritmo ricorsivo (de Boor e Cox)
7) **Si valutano le derivate delle funzioni base**

Fase 4

8) Si risolve il sistema e si determinano le spline di spostamento, velocità, accelerazione e jerk.

NURBS: Confronto funzioni base di ordine *k*





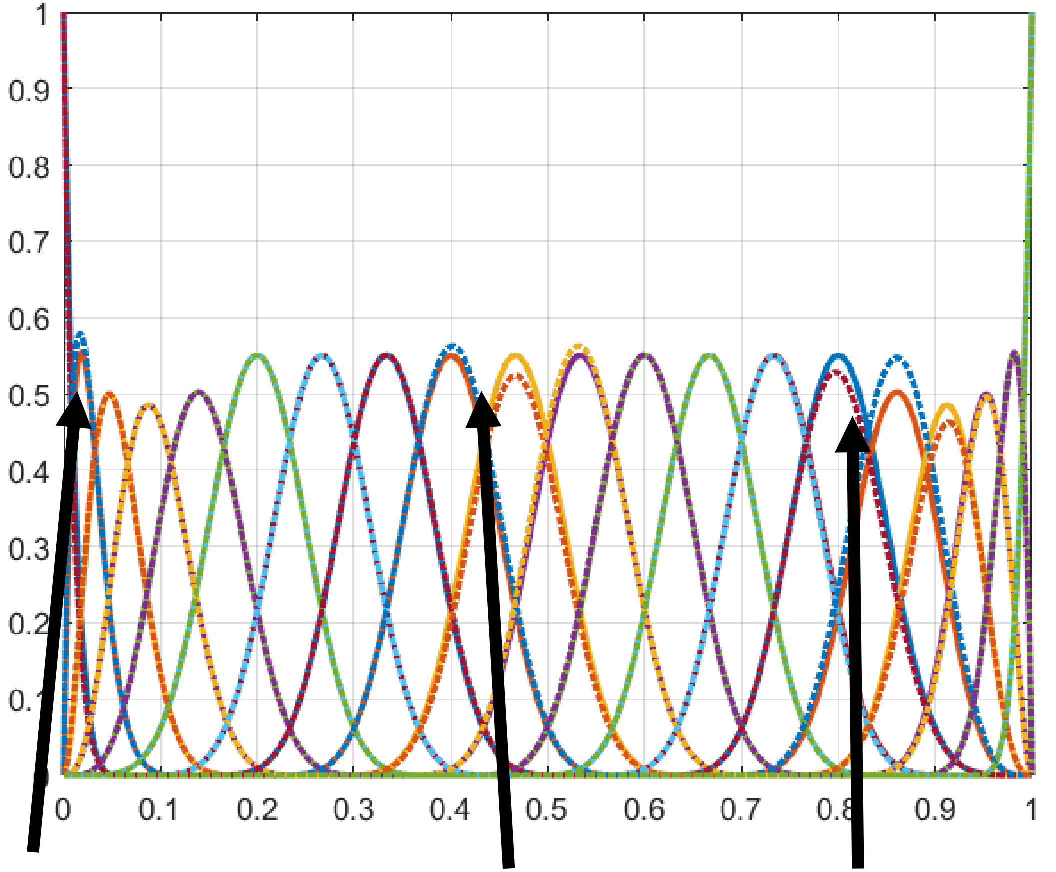
$$W = [2,1,1,1,1,1,1,1];$$

NURBS: Confronto funzioni base di ordine *k*

Esempio con elevato numero di vincoli:

$n = 20$

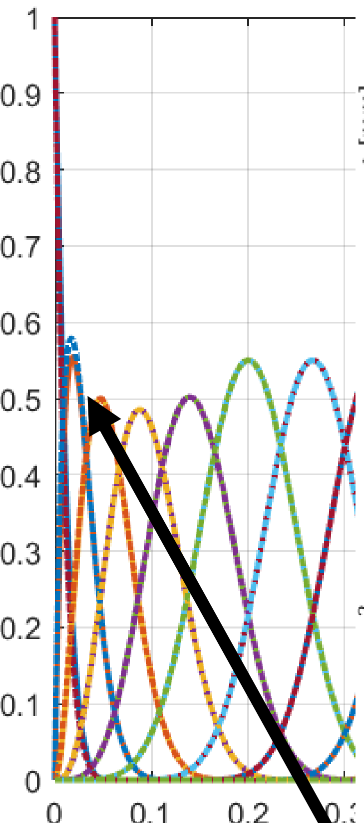
$k = 6$



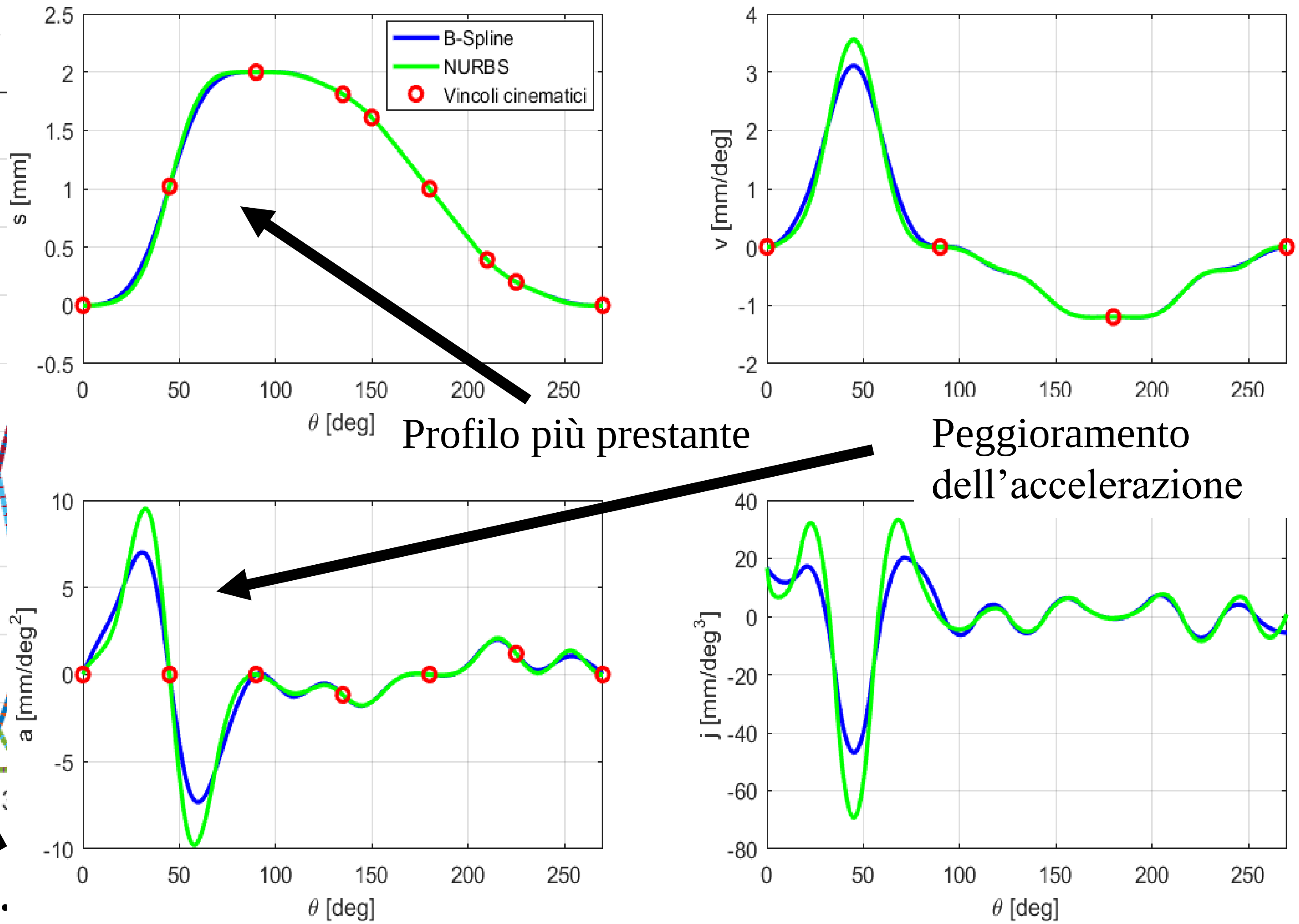
w = [0.8 1 1 1 1 1 1 1 1 0.9 1 1 1 1 1 1.2 1 1 1 1];

NURBS: Confronto funzioni base di

Esempio con



w = [0.



Profilo più prestante

Peggioramento dell’accelerazione

Ottimizzazione di profili