

TD3 - Compte-rendu

Clément Gaspard, Axel Murlanne, Nicolas Rodrigues

Trajectoires en dimension 1

Sauf mention contraire, tous les exemples présentés ci-dessous utilisent les données du fichier *cubic_zero_derivative_spline_example.json*, c'est-à-dire les points de passage suivants :

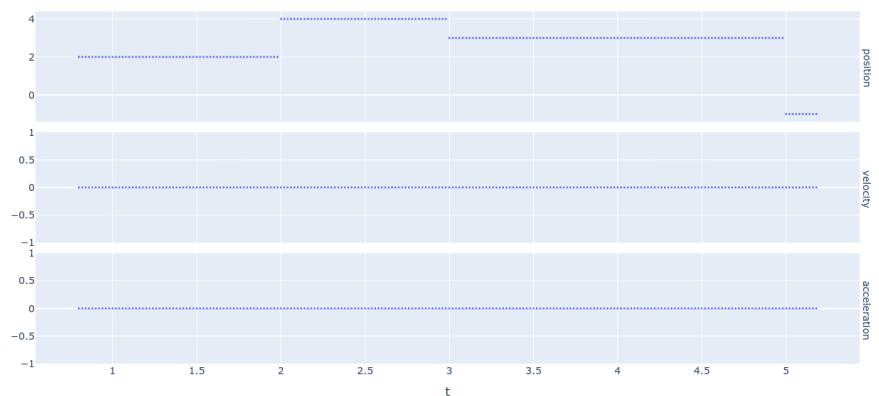
t	0	1	2	4	6
x	2	8	3	-1	2

(La valeur de *start* est 1, ce qui signifie que pour chaque point, on décale t de 1).

Constant Spline

La position est constante entre chaque point de passage, ce qui fait qu'elle n'est pas continue. Cette trajectoire n'appartient donc à aucune classe de continuité.

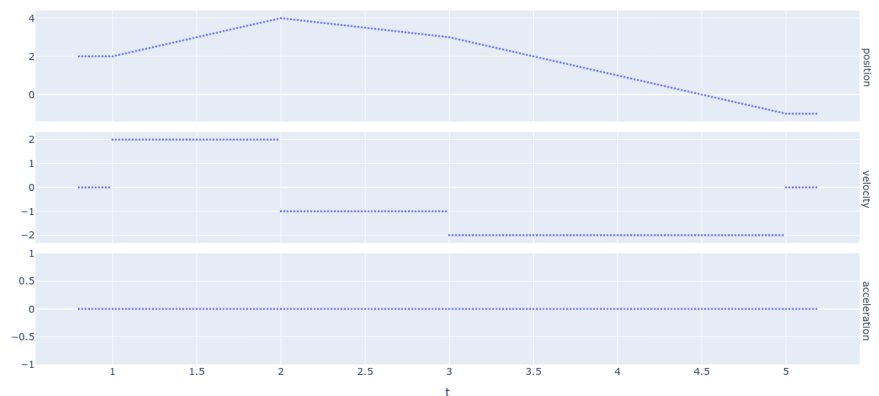
Dans cet exemple, on utilise les données du fichier *constant_example.json*.



Linear Spline

La position est linéaire entre chaque point de passage. La position est continue mais pas la vitesse, cette trajectoire appartient donc à la classe $C^0[t_0, t_n]$.

Dans cet exemple, on utilise les données du fichier *linear_example.json*.



Cubic Zero Derivative Spline

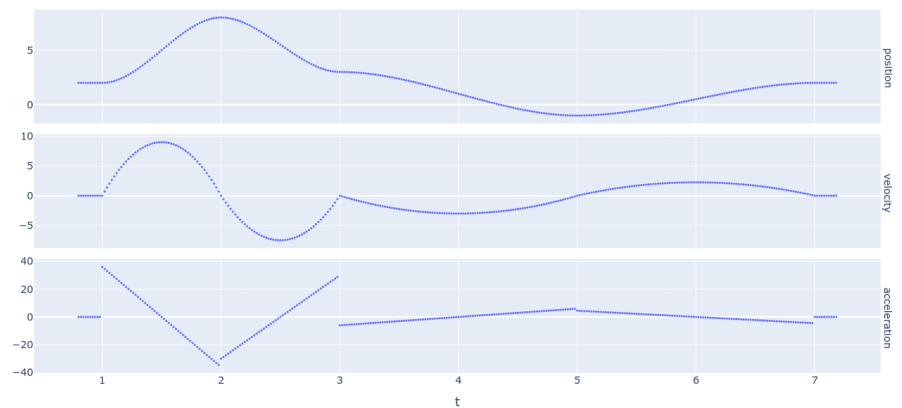
Ici, la trajectoire entre 2 points est cubique, et sur chaque point de passage la vitesse doit valoir 0.

Cette trajectoire est $C^1[t_0, t_n]$. En effet, entre chaque point de passage on a une trajectoire cubique, donc la vitesse est une fonction du second degré (et donc continue) ; et sur les points de passage, la vitesse est égale à 0,

donc la vitesse à la fin d'une trajectoire est la même que la vitesse au début de la suivante.

Ainsi, la vitesse est continue.

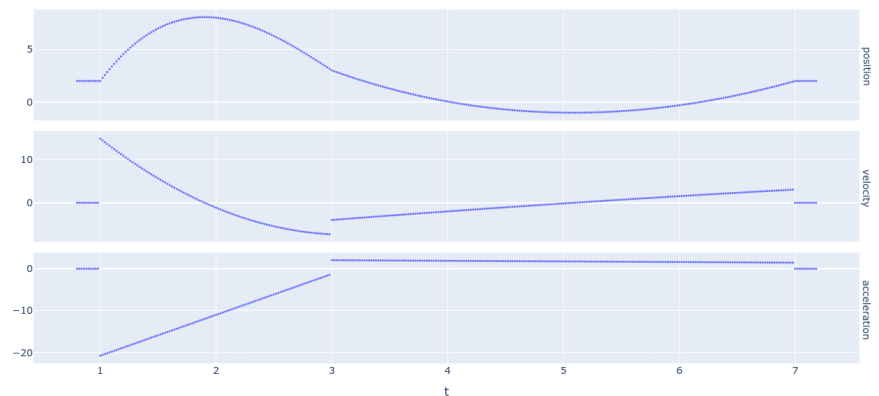
En revanche, il n'y a pas de contrainte sur l'accélération, celle-ci n'est pas forcément continue.



Cubic Wide Stencil Spline

Ici, la trajectoire entre 2 points est cubique, et elle prend en compte 4 points au lieu de 2.

Puisqu'on prend en compte 4 points, on pourrait s'attendre à ce que la trajectoire soit physiquement possible, c'est-à-dire avoir une vitesse et une accélération continue. Mais comme visible dans l'exemple ci-joint, ce n'est pas le cas. C'est une trajectoire $C^0[t_0, t_n]$.

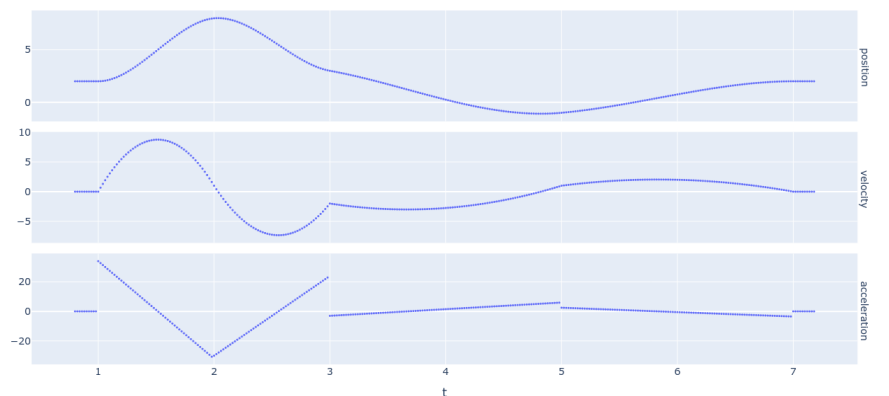


Cubic Custom Derivative Spline

Ici, la trajectoire entre 2 points est cubique, et sur chaque point de passage on donne aussi une information de vitesse.

Cette trajectoire est donc similaire au Cubic Zero Derivative Spline, elle appartient à la classe $C^1[t_0, t_n]$.

Dans cet exemple, on utilise les données du fichier *cubic_custom_derivative_spline.json*.

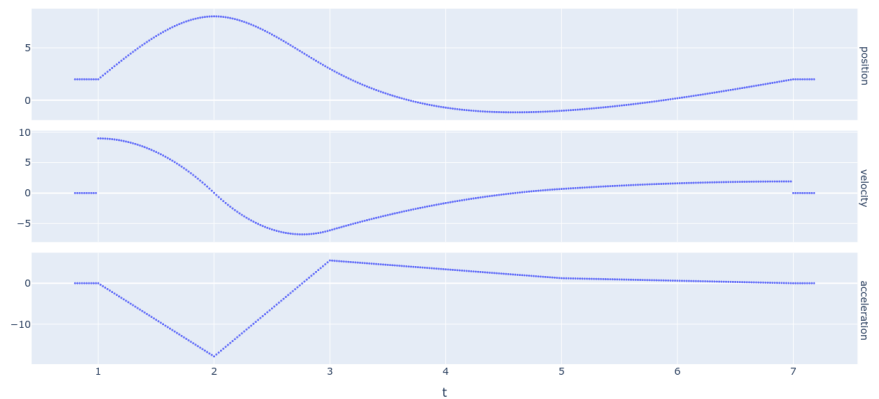


Natural Cubic Spline

C'est une spline cubique globale, c'est-à-dire que les trajectoires sont calculées globalement, et non pas couple de points par couple de points.

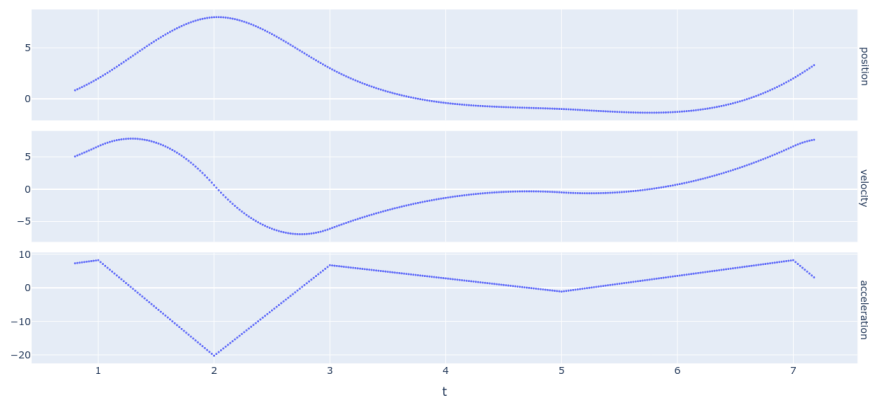
L'une des contraintes des splines cubiques globales est que la vitesse et l'accélération doivent être continues entre t_1 et t_{n-1} . Il n'y a en revanche aucune contrainte sur le jerk, donc la trajectoire est $C^2[t_0, t_n]$.

A noter qu'une contrainte spécifique aux splines naturelles est que l'accélération en 0 et en n sont nulles.



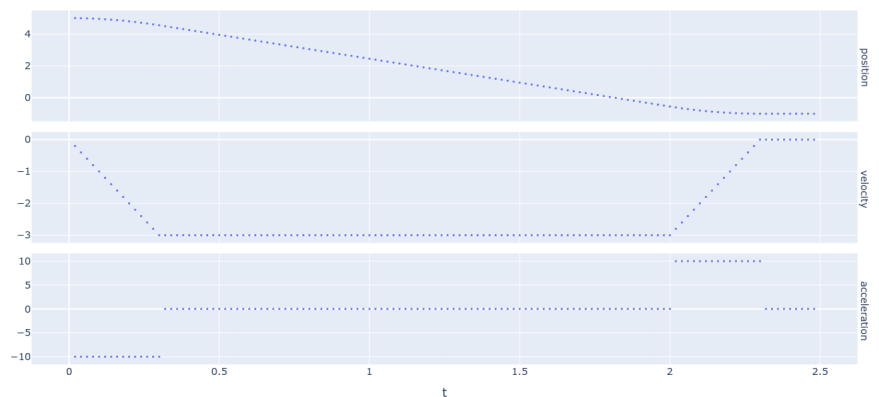
Periodic Cubic Spline

Le principe est le même que pour la Natural Cubic Spline, mais cette fois une contrainte spécifique est que la vitesse en 0 et en n sont égales. Le résultat est une trajectoire périodique, et donc une trajectoire qui se répète à l'infini. Ainsi, la classe de continuité est $C^2[t_0, t_{\text{infini}}]$.



Loi trapézoïdale en vitesse

Ici, on n'a pas d'information de temps, le but est simplement d'aller d'un point A à un point B le plus vite possible, sans dépasser une vitesse et une accélération maximales. Il n'y a en revanche pas de jerk maximal, ce qui fait que l'accélération n'est pas continue. La trajectoire est donc $C^1[t_0, t_n]$.



Dans cet exemple, on utilise les données du fichier *trapezoidal_lax_example.json*.

Trajectoires multidimensionnelles

Dans le cas des trajectoires multidimensionnelles, nous avons réussi à calculer la position et la vitesse, et à calculer l'accélération dans le cas où l'espace dans lequel les cibles sont spécifiées est le même que l'espace dans lequel on souhaite générer la trajectoire. En revanche, lorsque ces deux espaces sont différents, nous n'avons pas réussi à calculer l'accélération.

Voici un exemple des positions, vitesses et accélérations calculées en utilisant le fichier *rrr_zero_derivative_joint.json* :

