Programowanie animacji

Krzysztof Gdawiec

UNIWERSYTET ŚLĄSKI
INSTYTUT INFORMATYKI

Interpolacja

Interpolacja jest podstawą znacznej części animacji komputerowej.

Jednym z najprostszych jej przykładów jest interpolacja położenia obiektu w przestrzeni. Zrobienie tego poprawnie jest nietrywialne i wymaga uwzględnienia kilku zagadnień. Są to:

- dobór odpowiedniej funkcji interpolacji,
- parametryzacja funkcji na podstawie przebytej odległości,
- utrzymanie wymaganej kontroli nad położeniem punktu w zależności od czasu.

Często animator podaje listę wartości ustalonego parametru dla wybranych klatek (tzw. klatek kluczowych) animacji. Podstawowe pytanie brzmi, jaki jest najlepszy sposób wyznaczenia wartości tego parametru dla pozostałych klatek. Interpolowany parametr może być: położeniem obiektu, kątem nachylenia ramienia robota, przeźroczystością obiektu itp.

Podstawowe pojęcia

Krzywe mogą być zdefiniowane za pomocą wzorów na trzy sposoby:

- jawny,
- niejawny (uwikłany),
- parametryczny.

Podstawowe pojęcia

Krzywe mogą być zdefiniowane za pomocą wzorów na trzy sposoby:

- jawny,
- niejawny (uwikłany),
- parametryczny.

W jawnym sposobie krzywa dana jest poprzez wzór:

$$y = f(x).$$

Sposób ten umożliwia wygodne generowanie punktów przez podstawienie konkretnych wartości x jako argumentu funkcji f i obliczenie jej wartości. Wadą jest zależność otrzymanej krzywej od wyboru układu współrzędnych oraz nie można w ten sposób otrzymać krzywych mających więcej niż jeden punkt o tej samej współrzędnej x.

Sposób niejawny polega na podaniu równania postaci:

$$f(x, y) = 0.$$

Mając równanie tego typu możemy sprawdzić czy pewien punkt leży na krzywej, badając czy jego współrzędne x, y spełniają to równanie. Wadą tego sposobu jest trudność konstruowania punktów krzywej, a w praktyce często zachodzi taka potrzeba.

Sposób parametryczny polega na użyciu wzorów:

$$x = f(t),$$
$$y = g(t),$$

gdzie $f, g : [t_{min}, t_{max}] \to \mathbb{R}$ są funkcjami, a $t \in [t_{min}, t_{max}]$.

Podstawiając t do tych wzorów otrzymujemy odpowiadający punkt (x, y). Zatem konstruowanie punktów krzywej jest bardzo proste i wygodne.

Funkcje używane do określania krzywych mogą być klasyfikowane na podstawie definiujących je wyrażeń. Można tutaj wyróżnić funkcje:

- wielomianowe,
- wymierne,
- przestępne, np. funkcje trygonometryczne, logarytmiczne, wykładnicze itp.

Najczęściej używane krzywe w grafice komputerowej to wielomianowe krzywe trzeciego stopnia, nazywane krzywymi kubicznymi.

Ważnym pojęciem jest pojęcie ciągłości funkcji. Mówimy, że funkcja $f : \mathbb{R} \to \mathbb{R}$ jest ciągła w punkcie $x_0 \in \mathbb{R}$ jeśli

$$\lim_{x \to x_0} f(x) = f(x_0).$$

Ważnym pojęciem jest pojęcie ciągłości funkcji. Mówimy, że funkcja $f : \mathbb{R} \to \mathbb{R}$ jest ciągła w punkcie $x_0 \in \mathbb{R}$ jeśli

$$\lim_{x \to x_0} f(x) = f(x_0).$$

Jeśli funkcja jest ciągła to mówimy, że jest klasy C^0. Jeśli funkcja jest ciągła i dodatkowo ma ciągłą pochodną, to mówimy, że jest klasy C^1.

Ogólnie funkcja jest klasy C^n jeśli jest ciągła i ma ciągłe pochodne do rzędu n.

Ważnym pojęciem jest pojęcie ciągłości funkcji. Mówimy, że funkcja $f : \mathbb{R} \to \mathbb{R}$ jest ciągła w punkcie $x_0 \in \mathbb{R}$ jeśli

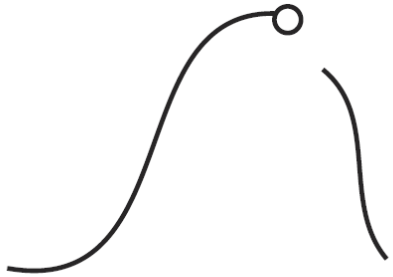
$$\lim_{x \to x_0} f(x) = f(x_0).$$

Jeśli funkcja jest ciągła to mówimy, że jest klasy C^0. Jeśli funkcja jest ciągła i dodatkowo ma ciągłą pochodną, to mówimy, że jest klasy C^1.

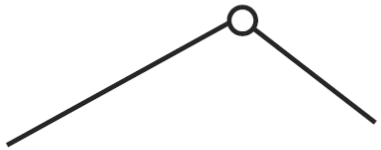
Ogólnie funkcja jest klasy C^n jeśli jest ciągła i ma ciągłe pochodne do rzędu n.

Jeśli funkcja jest określona kawałkami, tj. przy użyciu różnych wzorów dla argumentu z różnych fragmentów dziedziny, to nazywamy ją funkcją sklejaną i możemy mówić o lokalnych własnościach funkcji.

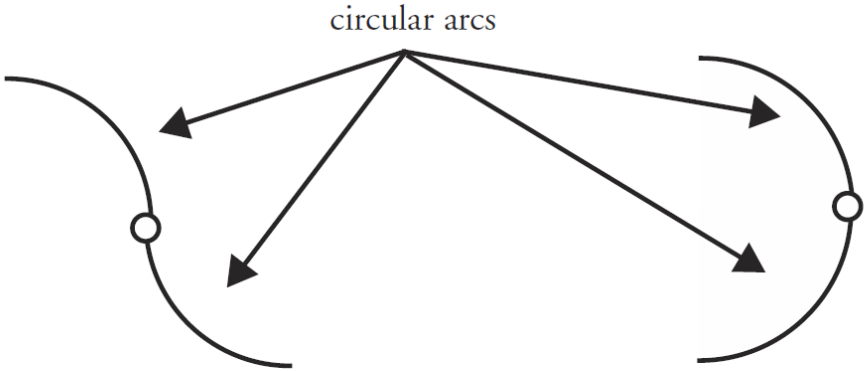
Istotnym zagadnieniem jest klasa ciągłości funkcji w punktach połączenia, jeśli wykresy funkcji opisujących ją kawałkami mają odpowiedni punkt wspólny.



Positional discontinuity at the point



Positional continuity but not tangential continuity at the point

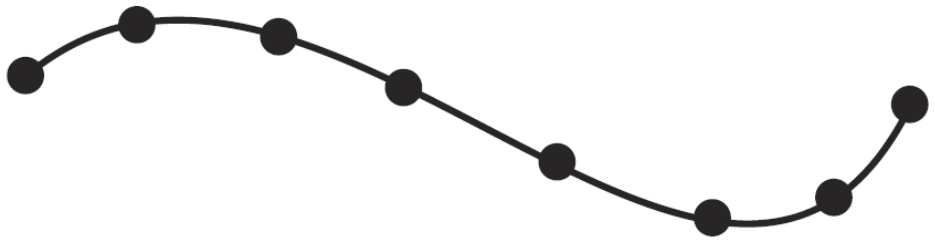


Positional and tangential continuity but not curvature continuity at the point

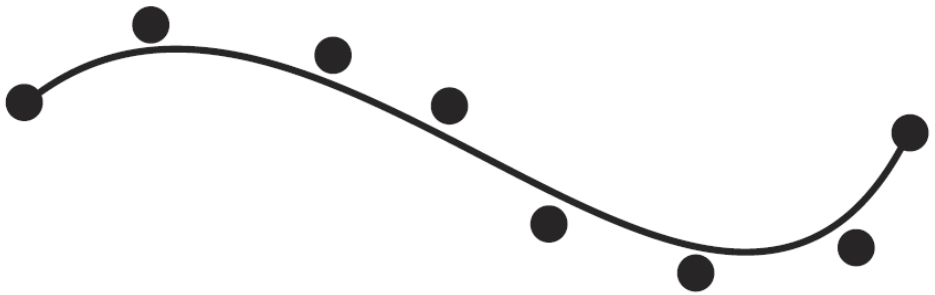
Positional, tangential, and curvature continuity at the point

Krzywe dzielimy również na:

- krzywe interpolacyjne – są to krzywe, które przechodzą przez ciąg danych punktów,



- krzywe aproksymacyjne – są to krzywe, które nie muszą przechodzić przez ciąg danych punktów, ale leżą w pobliżu tych punktów.



Interpolacja liniowa

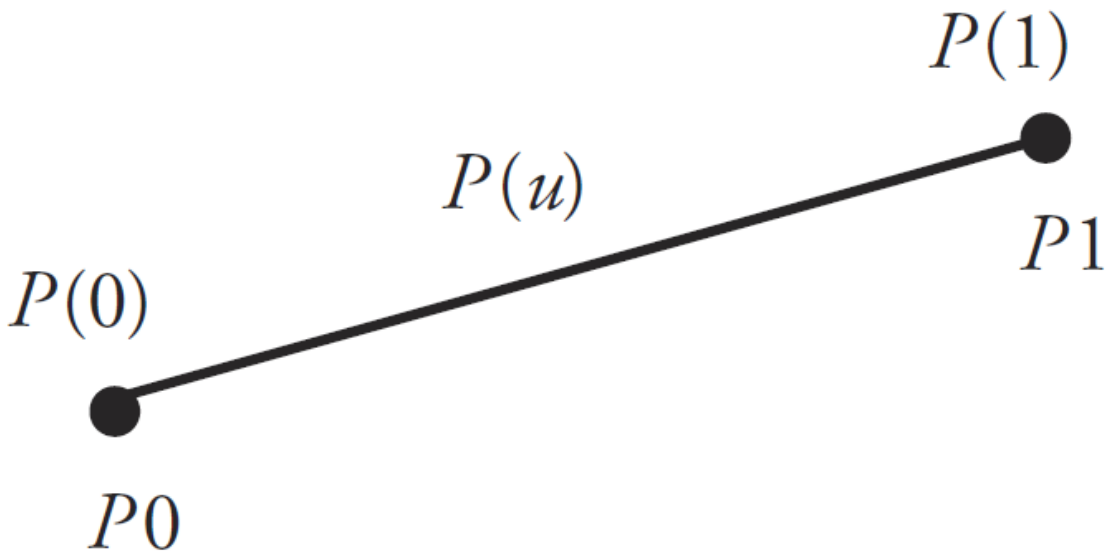
Interpolacja liniowa dwóch punktów P_0, P_1 dana jest wzorem:

$$P(u) = (1-u)P_0 + uP_1,$$

gdzie $u \in [0, 1]$.

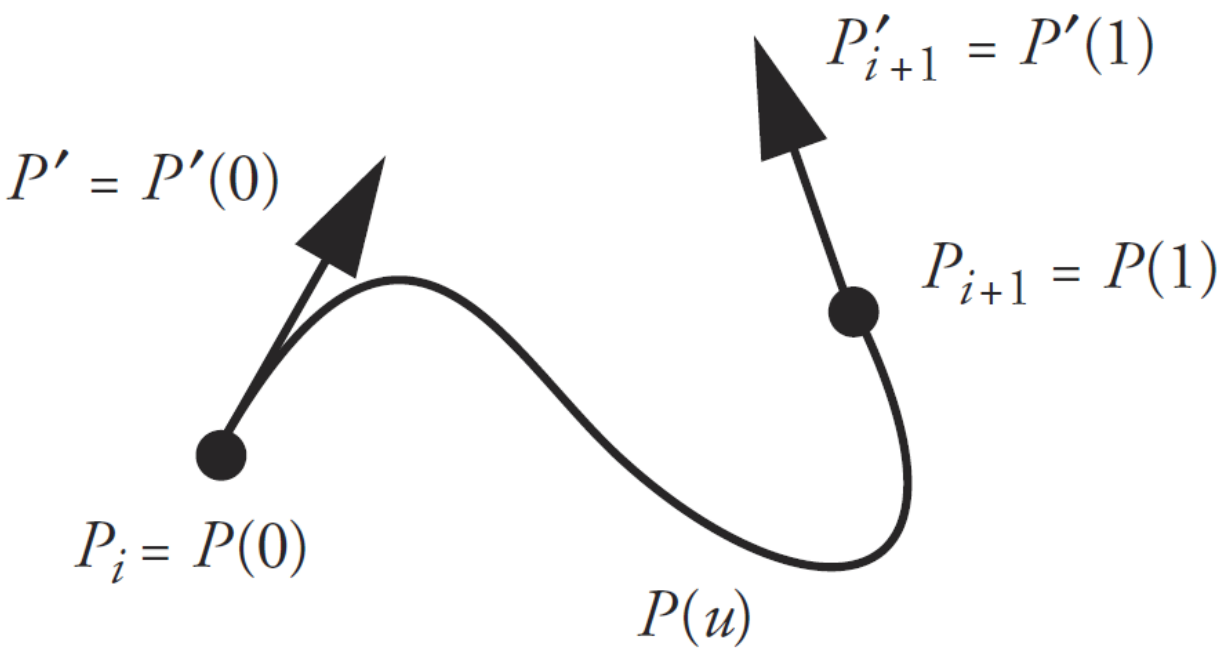
Wzór ten możemy też zapisać w postaci macierzowej:

$$P(u) = \begin{bmatrix} u & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \end{bmatrix} = U^T MB.$$


P(1)
P(u)
P1
P(0)
P0

Interpolacja Hermite'a

Krzywa interpolacyjna Hermite'a jest krzywą trzeciego stopnia łączącą dwa dane punkty. Oprócz tych punktów P_i, P_{i+1} należy określić wektory pochodnych parametryzacji P', P'_{i+1}.



Krzywą interpolacyjną Hermite'a możemy przedstawić w postaci macierzowej:

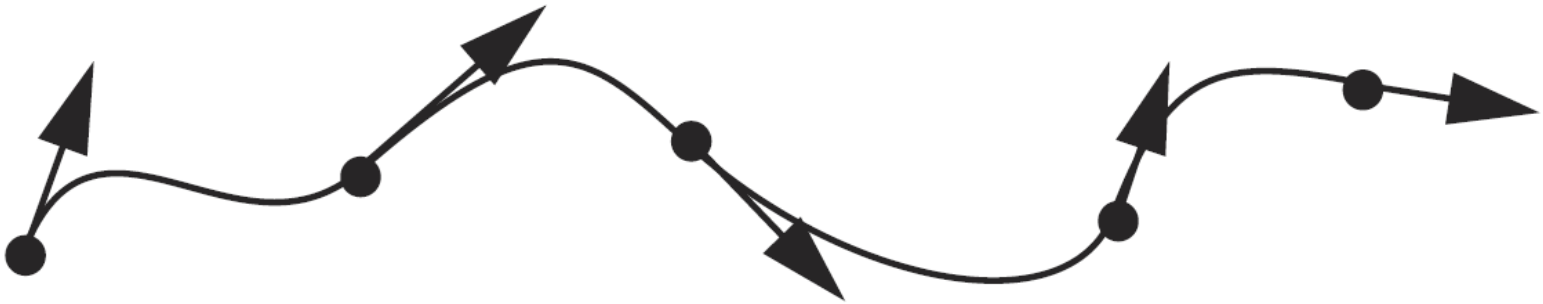
$$P(u) = U^T M B,$$

gdzie $U^T = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$,

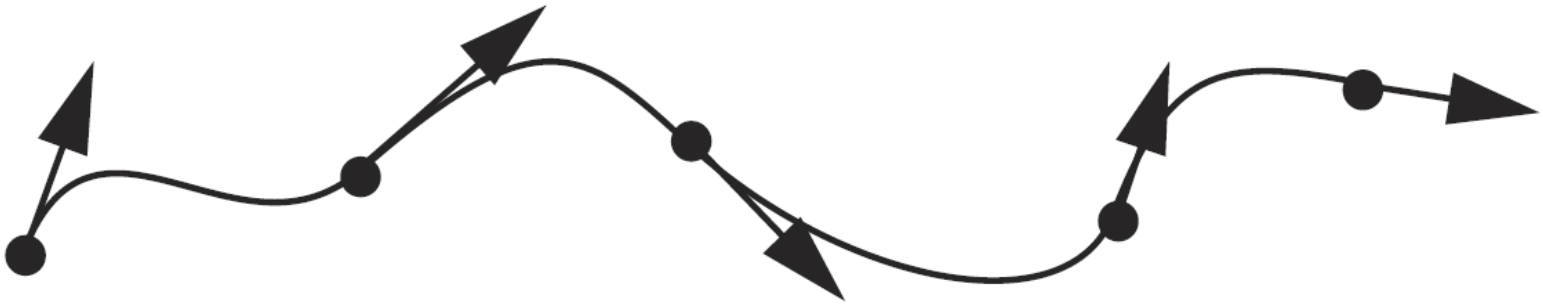
$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} P_i \\ P_{i+1} \\ P' \\ P'_{i+1} \end{bmatrix}.$$

Ciągłość stycznej krzywej zbudowanej z połączonych krzywych Hermite'a można zapewnić przez podanie wspólnych końców i wspólnych wektorów pochodnych.



Ciągłość stycznej krzywej zbudowanej z połączonych krzywych Hermite'a można zapewnić przez podanie wspólnych końców i wspólnych wektorów pochodnych.

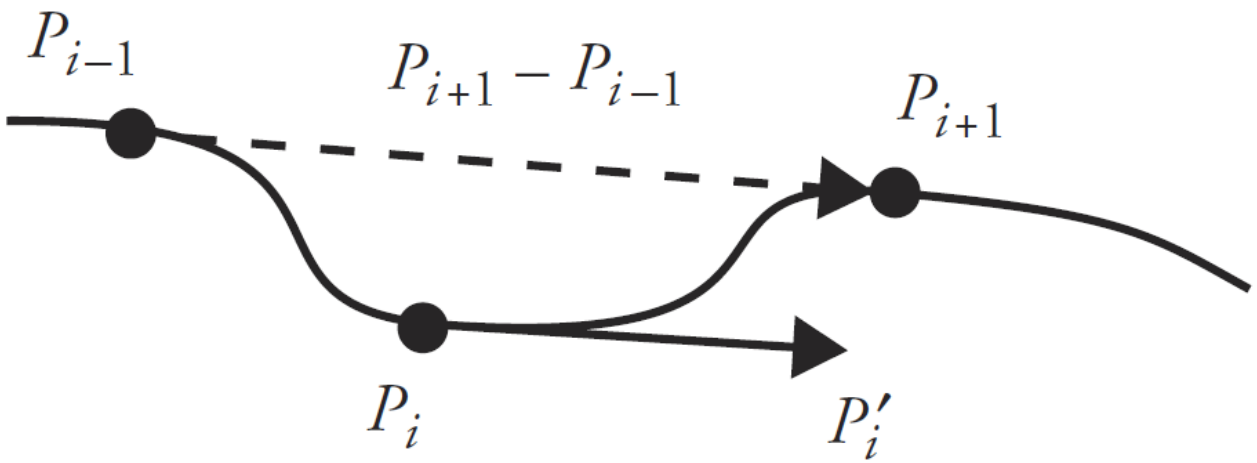


Konstrukcja takiej krzywej wymaga podania dla każdego punktu wektora pochodnej co może być zbyt pracochłonne. Istnieją różne sposoby uniknięcia tej pracy, np. użyciu pochodnych drugiego rzędu czy też sklejane krzywe Catmulla-Roma.

Krzywe sklejane Catmulla-Roma

Dla każdego punktu połączenia segmentów P_i wektor P_i' jest równy:

$$P_i' = \frac{1}{2}(P_{i+1} - P_{i-1}).$$



Macierze za pomocą których można przedstawić segment krzywej Catmulla-Roma między punktami P_i, P_{i+1} są następujące:

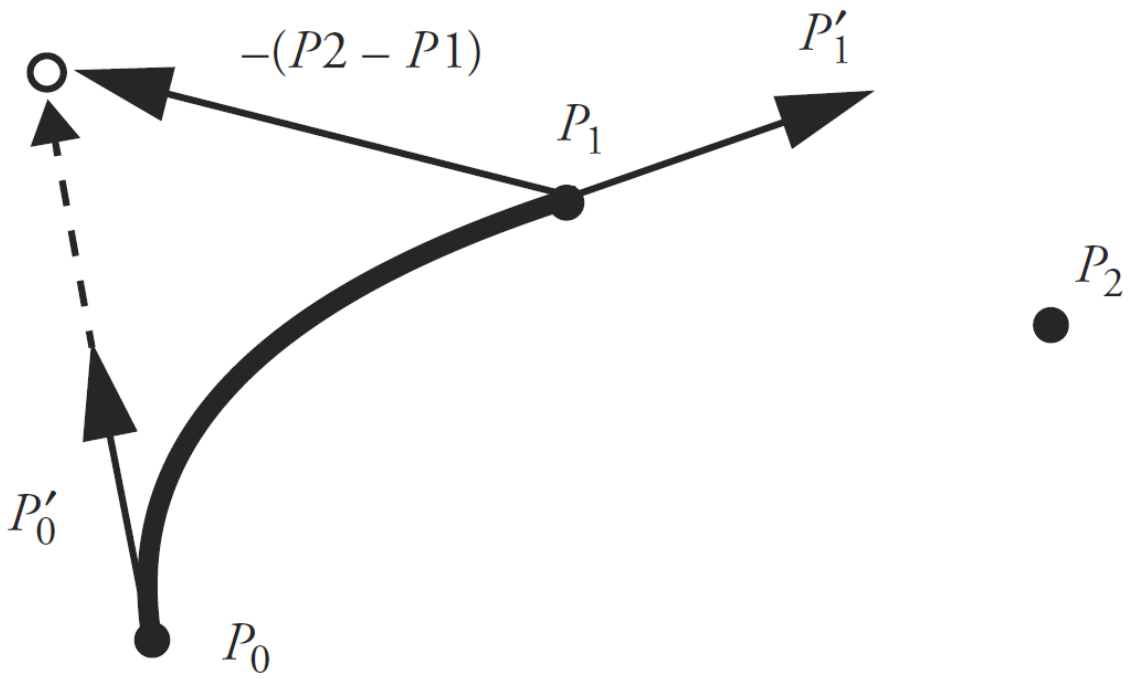
$$U^T = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix},$$

$$M = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}.$$

Wektory pochodnych dla punktu początkowego i końcowego krzywej może podać użytkownik. Istnieją też rozmaite konstrukcje automatycznego ich dobierania. Na przykład wektor pochodnej dla punktu początkowego P_0 możemy obliczyć ze wzoru:

$$P_0' = \frac{1}{2}(P_1 - (P_2 - P_1) - P_0) = \frac{1}{2}(2P_1 - P_2 - P_0).$$

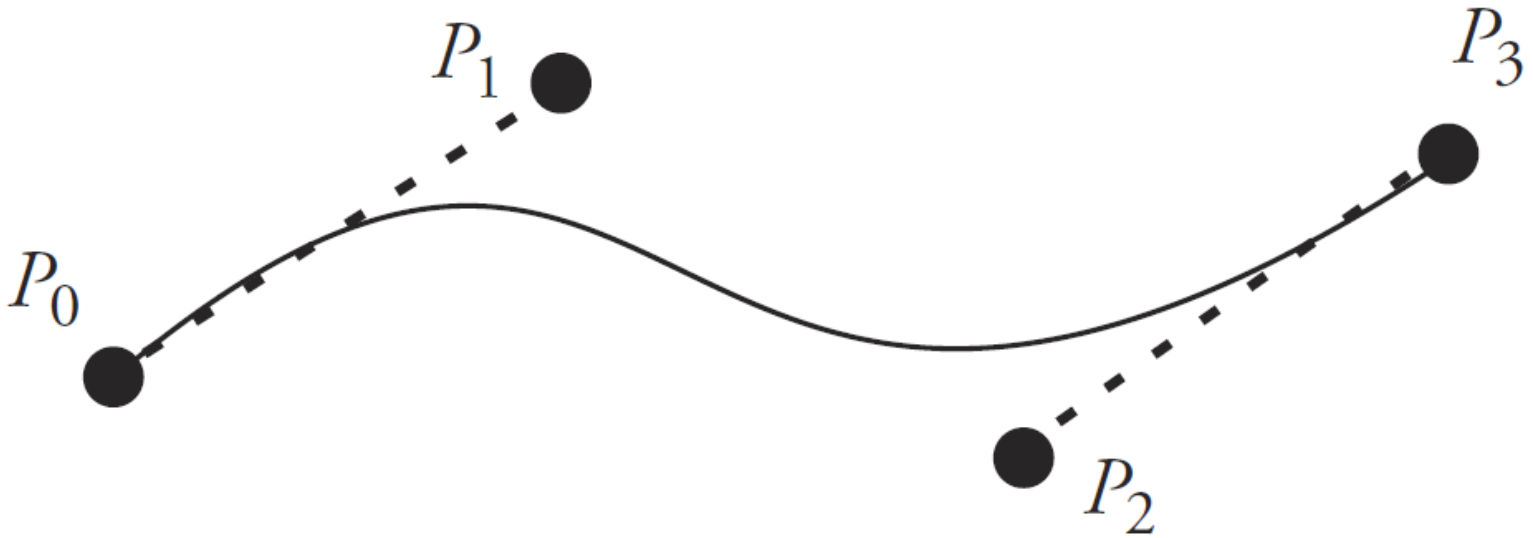


Wadą krzywych sklejanych Catmulla-Roma jest to, że wektor pochodnej odpowiadający punktowi połączenia segmentów P_i nie zależy od położenia tego punktu.

Zaletą jest ogromna prostota i szybkość obliczeń w ich konstrukcji.

Kubiczne krzywe Béziera

Kubiczna krzywa Béziera dana jest przez cztery punkty kontrolne P_0, P_1, P_2, P_3. Punkty P_0 i P_3 są interpolowane, zaś wektor pochodnej dla $u = 0$ jest równy $3(P_1 - P_0)$, a wektor dla $u = 1$ jest równy $3(p_3 - p_2)$.



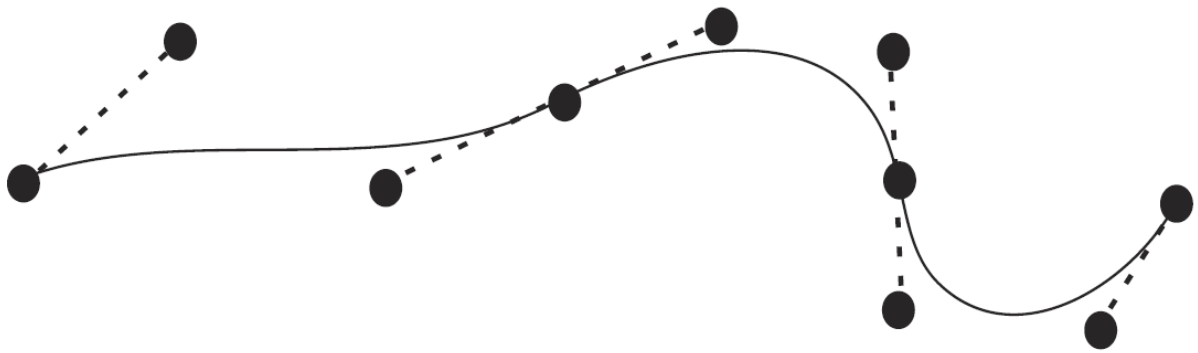
Macierz M za pomocą której można wyrazić krzywą ma postać:

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Macierz M za pomocą której można wyrazić krzywą ma postać:

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Gładkość połączenia segmentów krzywych Béziera można osiągnąć zapewniając współliniowość odpowiednich punktów kontrolnych.



Oprócz przedstawionych krzywych istnieje wiele innych rodzajów krzywych, np.

- krzywe kardynalne (szczególnym przypadkiem tych krzywych jest krzywa Catmulla-Roma),
- krzywe TCB lub Kochanka-Bartelsa,
- krzywe B-sklejane (B-splain),
- krzywe NURBS.

Sterowanie ruchem punktu wzdłuż krzywej

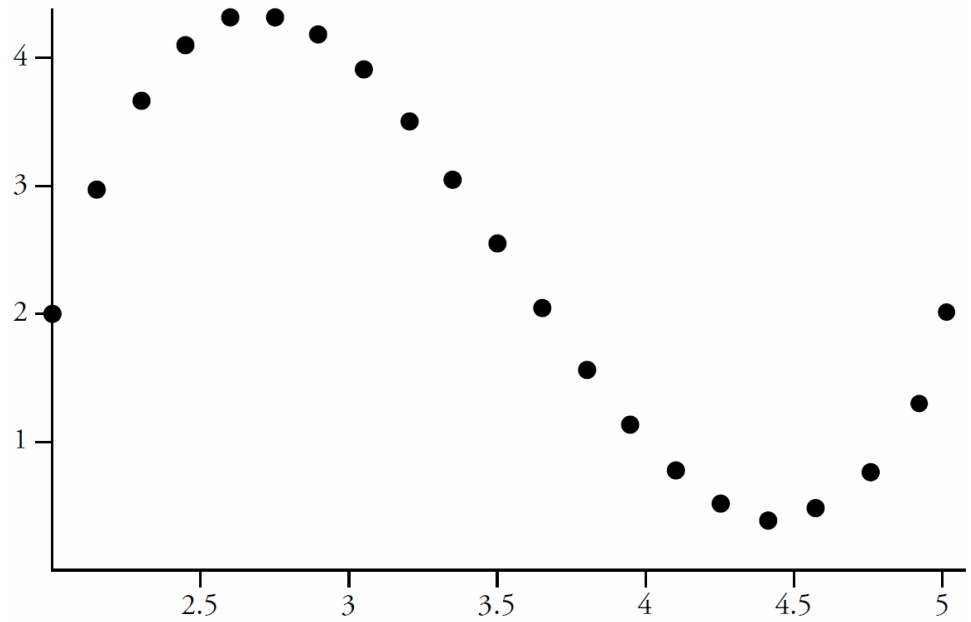
Pierwszym etapem prowadzącym do udostępnienia animatorowi kontroli nad ruchem jest ustalenie metody przechodzenia po krzywej równymi krokami. Gdy to już jest zrobione można udostępnić metody przyspieszania i spowalniania ruchu wzdłuż krzywej.

Sterowanie ruchem punktu wzdłuż krzywej

Pierwszym etapem prowadzącym do udostępnienia animatorowi kontroli nad ruchem jest ustalenie metody przechodzenia po krzywej równymi krokami. Gdy to już jest zrobione można udostępnić metody przyspieszania i spowalniania ruchu wzdłuż krzywej.

Załóżmy, że mamy daną funkcję $\mathrm{P} : [a, b] \to \mathbb{R}^3$. Współrzędne x, y, z punktów krzywej dla klatek kluczowych są określone przez użytkownika. Klatki te związane są z odpowiednimi wartościami parametru $u \in [a, b]$.

Zmienianie parametru (w naszym przypadku u) ze stałym krokiem nie oznacza, że kolejne wartości funkcji P leżą w tej samej odległości od siebie.



Jeśli zatem będziemy zmieniać u ze stałą szybkością, to nie oznacza to na ogół ruchu po krzywej ze stałą prędkością.

Aby zapewnić ruch po krzywej ze stałą prędkością należy znaleźć tzw. parametryzację łukową tej krzywej, taką że wartość parametru dla każdego punktu odpowiada długości łuku między tym punktem i pewnym ustalonym punktem.

Aby zapewnić ruch po krzywej ze stałą prędkością należy znaleźć tzw. parametryzację łukową tej krzywej, taką że wartość parametru dla każdego punktu odpowiada długości łuku między tym punktem i pewnym ustalonym punktem.

W większości zastosowań przybliżoną parametryzację łukową daje się znaleźć jedną z trzech metod:

- analityczne obliczanie długości łuku,
- gęste próbkowanie funkcji P,
- numeryczne całkowanie przy użyciu kwadratury Gaussa.

Krzywą przestrzeną będziemy nazywać kształt krzywej określającej tor ruchu, zaś funkcją odległości względem czasu nazywać będziemy funkcję wiążącą czas z przebytą odległością.

Krzywą przestrzeną będziemy nazywać kształt krzywej określającej tor ruchu, zaś funkcją odległości względem czasu nazywać będziemy funkcję wiążącą czas z przebytą odległością.

Potrzebować będziemy funkcji, która będzie wiązać czas z punktem krzywej przestrzennej. Użytkownik będzie określał funkcję odległości przebytej wzdłuż krzywej względem czasu.

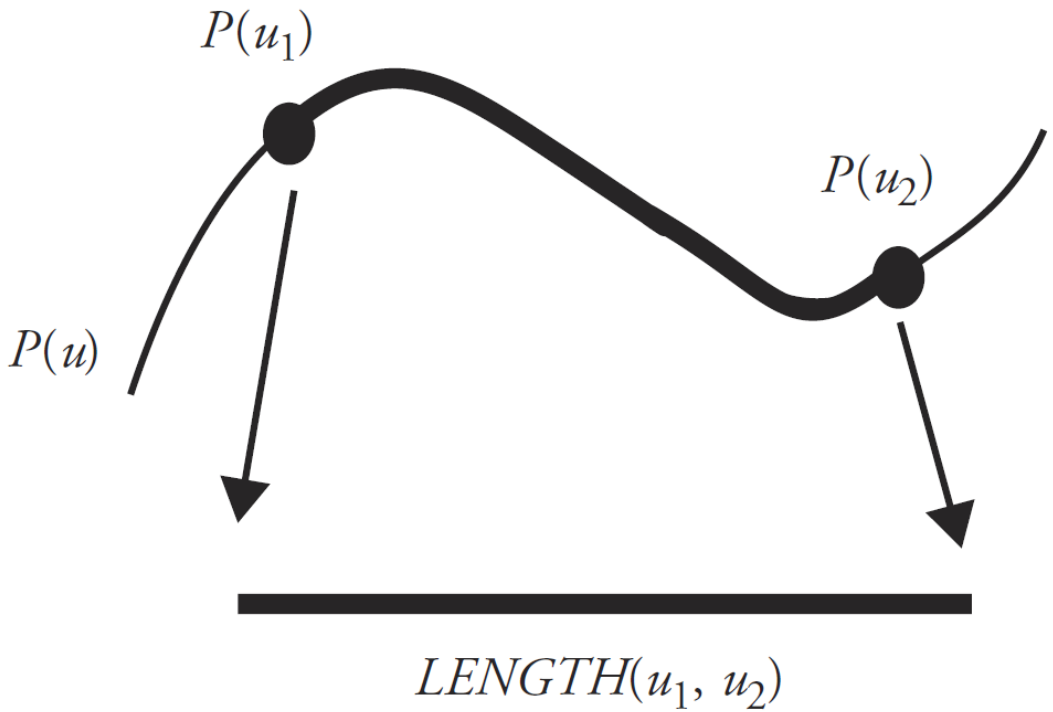
Tą odległość nazywać będziemy długością łuku i oznaczać będziemy przez s. Jeśli długość łuku jest obliczana w zależności od zmiennej u, to będziemy pisać $s = S(u)$.

Funkcję, której argumentem jest długość łuku, a wartością parametr u będziemy oznaczać przez U, a zatem $u = U(s)$.

Funkcja interpolacyjna wiąże wartość parametru u z punktem krzywej przestrzennej. Należy ustalić zależność między długością łuku a parametrem krzywej określoną przez parametryzację łukową tej krzywej.

Zależność ta zwykle jest nieliniowa. W szczególnym przypadku, gdy jednostkowa zmiana parametru odpowiada jednostkowemu przyrostowi długości łuku mówimy, że krzywa jest sparametryzowana długością łuku.

Niech $LENGTH(u_1, u_2)$ oznacza długość łuku wzdłuż krzywej przestrzennej między punktami $P(u_1)$, $P(u_2)$.



Mamy do rozwiązania dwa problemy:

- mając dane u_1 i u_2 chcemy obliczyć $LENGTH(u_1, u_2)$,
- mając daną długość łuku s i wartość parametru u_1 chcemy znaleźć liczbę u_2 taką, że $LENGTH(u_1, u_2) = s$. Jest to równoważne rozwiązaniu równania

$$s - LENGTH(u_1, u_2) = 0.$$

W ogólności żaden z tych problemów nie ma rozwiązania analitycznego, trzeba zatem do ich rozwiązania użyć metod numerycznych.

Pierwszym krokiem w sterowaniu prędkością ruchu wzdłuż krzywej jest ustalenie zależności między wartościami parametru i długością łuku.

Możemy tego dokonać przez określenie funkcji S, której wartością dla dowolnego u jest długość łuku od punktu początkowego do punktu odpowiadającego u. Jeśli wtedy można obliczyć (lub oszacować) odwrotność $u = S^{-1}(s) = U(s)$, to krzywą można efektywnie sparametryzować długością łuku, tzn. skonstruować parametryzację $\mathrm{P}(U(s))$.

Analityczne obliczanie długości łuku

Długość łuku krzywej od punktu $\mathrm{P}(u_1)$ do $\mathrm{P}(u_2)$ można znaleźć obliczając całkę:

$$s = \int_{u_1}^{u_2} \left| \frac{\mathrm{dP}}{\mathrm{d}u} \right| \mathrm{d}u,$$

gdzie

$$\frac{\mathrm{dP}}{\mathrm{d}u} = \left(\frac{\mathrm{d}x(u)}{\mathrm{d}u}, \frac{\mathrm{d}y(u)}{\mathrm{d}u}, \frac{\mathrm{d}z(u)}{\mathrm{d}u} \right)$$

oraz

$$\left| \frac{\mathrm{dP}}{\mathrm{d}u} \right| = \sqrt{\left(\frac{\mathrm{d}x(u)}{\mathrm{d}u} \right)^2 + \left(\frac{\mathrm{d}y(u)}{\mathrm{d}u} \right)^2 + \left(\frac{\mathrm{d}z(u)}{\mathrm{d}u} \right)^2}.$$

Analityczne obliczanie długości łuku

Długość łuku krzywej od punktu $\mathrm{P}(u_1)$ do $\mathrm{P}(u_2)$ można znaleźć obliczając całkę:

$$s = \int_{u_1}^{u_2} \left| \frac{\mathrm{dP}}{\mathrm{d}u} \right| \mathrm{d}u,$$

gdzie

$$\frac{\mathrm{dP}}{\mathrm{d}u} = \left(\frac{\mathrm{d}x(u)}{\mathrm{d}u}, \frac{\mathrm{d}y(u)}{\mathrm{d}u}, \frac{\mathrm{d}z(u)}{\mathrm{d}u} \right)$$

oraz

$$\left| \frac{\mathrm{dP}}{\mathrm{d}u} \right| = \sqrt{\left(\frac{\mathrm{d}x(u)}{\mathrm{d}u} \right)^2 + \left(\frac{\mathrm{d}y(u)}{\mathrm{d}u} \right)^2 + \left(\frac{\mathrm{d}z(u)}{\mathrm{d}u} \right)^2}.$$

Niestety analityczne znalezienie parametryzacji łukowej jest niemożliwe dla wielu użytecznych rodzajów krzywych ze względu na trudność odwrócenia funkcji górnej granicy całkowania.

Szacowanie długości łuku przez różnice progresywne

Jest to najprostsza metoda ustalania zależności między wartością parametru a długością łuku. Wygląda ona następująco:

- tablicujemy krzywą dla wielu wartości parametru,
- obliczamy odległości pomiędzy kolejnymi dwoma punktami (przybliżamy długości łuków pomiędzy tymi punktami),
- budujemy tablicę długości łuków indeksowaną wartościami parametru.

Na przykład mając daną krzywą P obliczamy punkty krzywej dla $u = 0.0, 0.05, 0.1, \ldots, 1.0$. Tablica z zapamiętanymi zsumowanymi odległościami jest oznaczona symbolem G, a jej elementy obliczane są następująco:

$$\begin{aligned}
G[0] &= 0.0, \\
G[1] &= \rho(\mathrm{P}(0.0), \mathrm{P}(0.05)), \\
G[2] &= G[1] + \rho(\mathrm{P}(0.05), \mathrm{P}(0.1)), \\
G[3] &= G[2] + \rho(\mathrm{P}(0.1), \mathrm{P}(0.15)), \\
&\ldots \\
G[20] &= G[19] + \rho(\mathrm{P}(0.95), \mathrm{P}(1.0)),
\end{aligned}$$

gdzie $\rho(\mathrm{P}_1, \mathrm{P}_2)$ jest odległością P_1 od P_2.

Chcemy poznać odległość od początku krzywej do punktu odpowiadającego $u = 0.73$.

Szukamy elementu w tablicy najbliższemu 0.73. Ponieważ wartości parametru są równoodległe, więc możemy to policzyć ze wzoru:

$$i = \left\lfloor \frac{u}{d} + 0.5 \right\rfloor = \left\lfloor \frac{0.73}{0.05} + 0.5 \right\rfloor = 15.$$

Zgrubne oszacowanie wynosi 0.959.

Lepsze oszacowanie można otrzymać przez interpolację liniową wartości po obu stronach wartości parametru.

Index	Parametric Entry	Arc Length (G)
0	0.00	0.000
1	0.05	0.080
2	0.10	0.150
3	0.15	0.230
4	0.20	0.320
5	0.25	0.400
6	0.30	0.500
7	0.35	0.600
8	0.40	0.720
9	0.45	0.800
10	0.50	0.860
11	0.55	0.900
12	0.60	0.920
13	0.65	0.932
14	0.70	0.944
15	0.75	0.959
16	0.80	0.972
17	0.85	0.984
18	0.90	0.994
19	0.95	0.998
20	1.00	1.000

Potrzebujemy indeks elementu o największej wartości parametru mniejszej niż 0.73:

$$i = \left\lfloor \frac{u}{d} \right\rfloor = \left\lfloor \frac{0.73}{0.05} \right\rfloor = 14.$$

Interpolujemy wartości:

$$s = G[i] + \frac{u - u[i]}{u[i+1] - u[i]}(G[i+1] - G[i])$$
$$= 0.944 + \frac{0.73 - 0.7}{0.75 - 0.7}(0.959 - 0.944) = 0.953$$

Index	Parametric Entry	Arc Length (G)
0	0.00	0.000
1	0.05	0.080
2	0.10	0.150
3	0.15	0.230
4	0.20	0.320
5	0.25	0.400
6	0.30	0.500
7	0.35	0.600
8	0.40	0.720
9	0.45	0.800
10	0.50	0.860
11	0.55	0.900
12	0.60	0.920
13	0.65	0.932
14	0.70	0.944
15	0.75	0.959
16	0.80	0.972
17	0.85	0.984
18	0.90	0.994
19	0.95	0.998
20	1.00	1.000

Rozwiązanie pierwszego z wymienionych wcześniej problemów można znaleźć wykonując takie obliczenie dwukrotnie i odejmując otrzymane wyniki.

Rozwiązanie pierwszego z wymienionych wcześniej problemów można znaleźć wykonując takie obliczenie dwukrotnie i odejmując otrzymane wyniki.

Jeśli musimy znaleźć wartość u na podstawie danej długości łuku, to należy znaleźć w tablicy element najbliższy danej długości łuku i użyć indeksu tego elementu do oszacowania wartości parametru.

Punkty na krzywej nie są położone w równych odległościach, więc musimy zastosować procedurę przeszukiwania tablicy. Wiemy, że ciąg liczb w tablicy jest rosnący, więc możemy wykorzystać przeszukiwanie binarne.

Oszacowanie parametru możemy zrobić na podstawie znalezionego indeksu lub użyć interpolacji liniowej.

Rozwiązanie drugiego problemu można otrzymać przez użycie tablicy do oszacowania długości łuku od początku do punktu odpowiadającego u_1, dodanie jej do s i ponowne użycie tablicy do oszacowania wartości parametru.

Rozwiązanie drugiego problemu można otrzymać przez użycie tablicy do oszacowania długości łuku od początku do punktu odpowiadającego u_1, dodanie jej do s i ponowne użycie tablicy do oszacowania wartości parametru.

Zaletami różnic progresywnych są: łatwość implementacji, intuicyjność, duża szybkość obliczeń.

Wadą jest fakt, że szacowanie długości łuku i interpolacja liniowa wprowadzają błędy do obliczeń. Błędy można zmniejszyć na kilka sposobów:

- gęstsze tablicowanie,
- lepsze metody interpolacji,
- adaptacyjne obliczanie różnic progresywnych.

Adaptacyjne obliczanie różnic progresywnych

Metoda ta polega na zastosowaniu do rozpatrywanego łuku krzywej testu sprawdzającego czy oszacowanie długości łuku mieści się w granicach przyjętej tolerancji.

Test polega na porównaniu tego oszacowania z sumą oszacowań długości dwóch połówek danego łuku. Jeśli różnica przekracza dopuszczalny próg tolerancji, to te dwie połowy łuku są wstawiane do listy łuków do sprawdzenia. Na każdym poziomie rekurencyjnego podziału łuku próg tolerancji jest zmniejszany dwukrotnie. Jeśli różnica nie przekracza progu, to bieżące oszacowanie jest przyjmowane za dostatecznie dokładne i jest używane w dalszych obliczeniach.

Początkowo łuk jest całą krzywą.

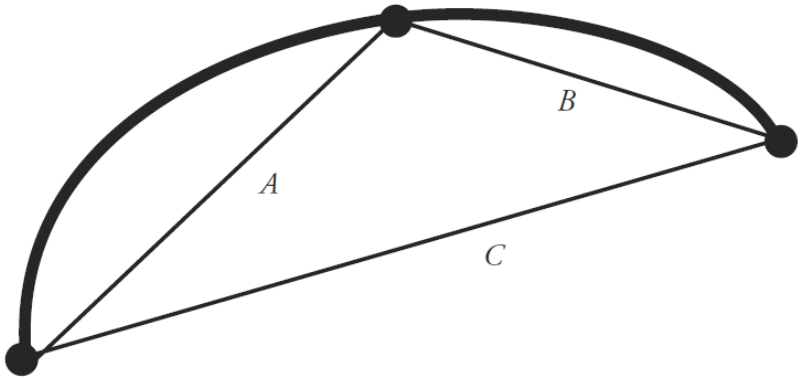
Na początku w tablicy umieszczamy element odpowiadający punktowi początkowemu krzywej $(0, 0, \mathrm{P}(0))$ i umieszczamy na liście do sprawdzenia przedział $[0, 1]$.

Procedura działa na fragmentach wyjmowanych z listy i kończy się gdy lista jest pusta.

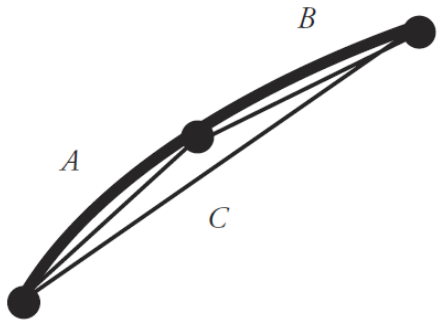
Bierzemy pierwszy element listy $[u_1, u_2]$ i obliczamy punkt środkowy łuku, tj. punkt odpowiadający środkowi przedziału zmienności parametru tego łuku. Ponadto obliczamy punkt końcowy łuku (punkt początkowy mamy w tablicy).

Za oszacowanie długości łuku i jego połówek przyjmujemy długości odcinków łączących te trzy punkty krzywej. Przeprowadzamy test

$$\|\mathrm{P}(u_1) - \mathrm{P}(u_2)\| - (\|\mathrm{P}(u_1) - \mathrm{P}(0.5(u_1 + u_2))\| + \|\mathrm{P}(0.5(u_1 + u_2)) - \mathrm{P}(u_2)\|) < \varepsilon$$



Lengths $A + B - C$ above error tolerance

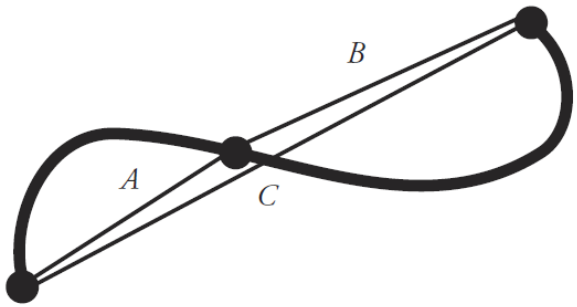


Lengths $A + B - C$ within error tolerance

Jeśli warunek nie jest spełniony, to obie połowy łuku są wstawiane do listy razem ze swoim progiem tolerancji błędu ($\varepsilon/2^n$).

Jeśli warunek jest spełniony, to zapamiętujemy w tablicy dwa nowe elementy odpowiadające łukom.

Pewien problem pojawia się w momencie gdy punkt środkowy łuku leży blisko odcinka łączącego jego końce.



Lengths $A + B - C$ erroneously report that the error is within tolerance

W takim przypadku warto wymusić podział co najmniej do pewnego poziomu niezależnie od wyników testu i zastosować adaptację na dalszych poziomach.

Ponieważ tablica jest otrzymana w sposób adaptacyjny nie jest możliwe bezpośrednie obliczanie indeksu elementu na podstawie wartości parametru.

Zależnie od reprezentacji danych może być potrzebne wyszukiwanie sekwencyjne albo binarne.

Po znalezieniu odpowiednich elementów, podobnie jak dla różnic progresywnych, do oszacowania długości łuku możemy użyć liczby wziętej z tablicy lub dokonać interpolacji liczb przechowywanych w sąsiednich elementach.

Szacowanie długości łuku przez całkowanie numeryczne

Obliczanie funkcji długości łuku polega na obliczeniu całki:

$$\int_{u_1}^{u_2} \left| \frac{\mathrm{d}\mathrm{P}}{\mathrm{d}u} \right| \mathrm{d}u.$$

Do jej obliczenia można wykorzystać metody numeryczne. W metodach tych przybliża się całkę sumą wartości funkcji podcałkowej w pewnych punktach tzw. węzłach kwadratury, pomnożonych przez odpowiednie współczynniki.

W metodach Simpsona i trapezów używane są wartości funkcji podcałkowej w węzłach równoodległych. Metoda kwadratury Gaussa jest oparta na węzłach nierównoodległych, dobranych tak, aby otrzymać możliwie dużą dokładność przy małej liczbie węzłów.

Kwadratura Gaussa jest zwykle zdefiniowana dla przedziału całkowania $[-1, 1]$. Wartości funkcji f obliczane są w ustalonych punktach u_i w tym przedziale i mnożone przez (obliczony wcześniej) współczynnik w_i:

$$\int_{-1}^{1} f(u)\,\mathrm{d}u = \sum_i w_i f(u_i).$$

Jeśli chcemy obliczyć całkę funkcji g w dowolnym przedziale $[a, b]$, to można zastąpić funkcję g przez $f(u) = g(h(u))$, gdzie

$$h(u) = \frac{u(b-a) + b + a}{2}$$

oraz $u \in [-1, 1]$.

Dzięki takiemu podstawieniu otrzymujemy:

$$\begin{aligned}\int_a^b g(t)\,\mathrm{d}t &= \int_{-1}^{1} g(h(u))h'(u)\,\mathrm{d}u \\ &= \int_{-1}^{1} \frac{b-a}{2} g\left(\frac{(b-a)u+b+a}{2}\right)\,\mathrm{d}u.\end{aligned}$$

Współczynniki i węzły kwadratur Gaussa różnych rzędów zostały obliczone i można je znaleźć w wielu podręcznikach metod numerycznych.

Sterowanie prędkością

Mając krzywą sparametryzowaną długością łuku możemy sterować prędkością ruchu wzdłuż niej.

Przebywanie krzywej krokami, którym odpowiada stała długość łuku daje w rezultacie ruch ze stałą prędkością.

Bardziej skomplikowane przypadki można otrzymać stosując funkcje sterowania prędkością, które wiążą stałe przyrosty nowego parametru, np. czasu, ze specjalnie dobieranymi długościami przebywanych łuków. Częsty przypadek polega na rozpędzeniu, a następnie spowolnieniu ruchu wzdłuż krzywej (ang. ease-in/ease-out).

Oznaczmy przez t czas, a przez s długość przebytego łuku.

W praktyce po znalezieniu parametryzacji łukowej modyfikujemy ją tak, aby parametr całej krzywej zmieniał się od 0 do 1. Jest to tak zwany znormalizowany parametr długości łuku.

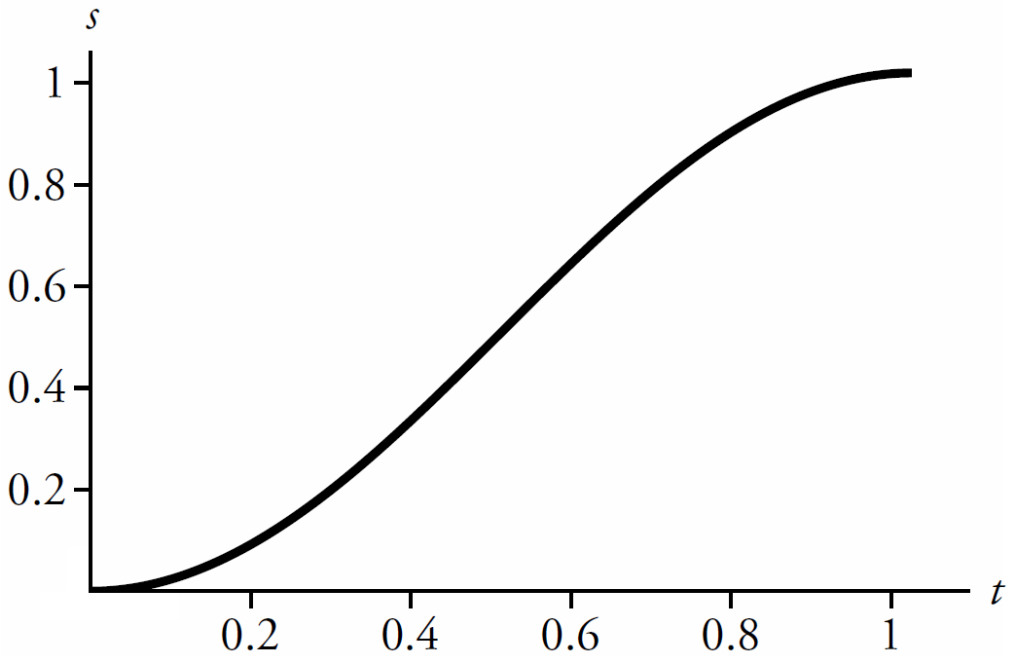
Oznaczmy przez t czas, a przez s długość przebytego łuku.

W praktyce po znalezieniu parametryzacji łukowej modyfikujemy ją tak, aby parametr całej krzywej zmieniał się od 0 do 1. Jest to tak zwany znormalizowany parametr długości łuku.

Prędkością ruchu wzdłuż krzywej można sterować przez złożenie parametryzacji łukowej z inną niż liniowa funkcją zmiennej t. Odwzorowanie czasu na długość łuku jest niezależne od kształtu samej krzywej, np. krzywa może być odcinkiem prostej, a zależność parametru długości łuku od czasu może być wielomianem trzeciego stopnia.

Początkowa wartość s równa jest 0, powoli rośnie i nabiera prędkości (która jest największa w środku przedziału), a następnie z coraz mniejszą prędkością dąży do 1.

Wartości zmiennej s są podstawiane w miejsce parametru długości łuku.



Naszym celem jest określenie, w sposób jawny lub nie, funkcji odległości od czasu $S(t)$, której wartość dla danego t jest długością drogi przebytej wzdłuż krzywej.

Krzywa w przestrzeni określa miejsce, w którym należy się znaleźć. Natomiast funkcja odległości określa kiedy ma to nastąpić.

Naszym celem jest określenie, w sposób jawny lub nie, funkcji odległości od czasu $S(t)$, której wartość dla danego t jest długością drogi przebytej wzdłuż krzywej.

Krzywa w przestrzeni określa miejsce, w którym należy się znaleźć. Natomiast funkcja odległości określa kiedy ma to nastąpić.

Długość łuku od początku krzywej do położenia osiągniętego w chwili t jest równa $S(t)$. Jeśli położenie określa przesunięcie obiektu, to przesuwa się on wzdłuż krzywej z prędkością proporcjonalną do $S'(t)$.

Tablica długości łuku może być użyta do znalezienia odpowiedniej wartości parametru $u = U(s)$ odpowiadającego tej długości łuku. Następnie obliczany jest punkt p krzywej odpowiadający u czyli $\mathrm{p} = \mathrm{P}(U(S(t)))$.

Najczęściej przyjmujemy pewne ograniczenia:

- funkcja odległości względem czasu powinna być monotoniczna, tj. ruch po krzywej powinien odbywać się bez cofania,
- funkcja odległości powinna być ciągła, tj. nie powinno być nagłych skoków między oddalonymi punktami krzywej,
- $S(0) = 0$ oraz $\mathit{calkowita\_odleglosc} = S(\mathit{calkowity\_czas})$.

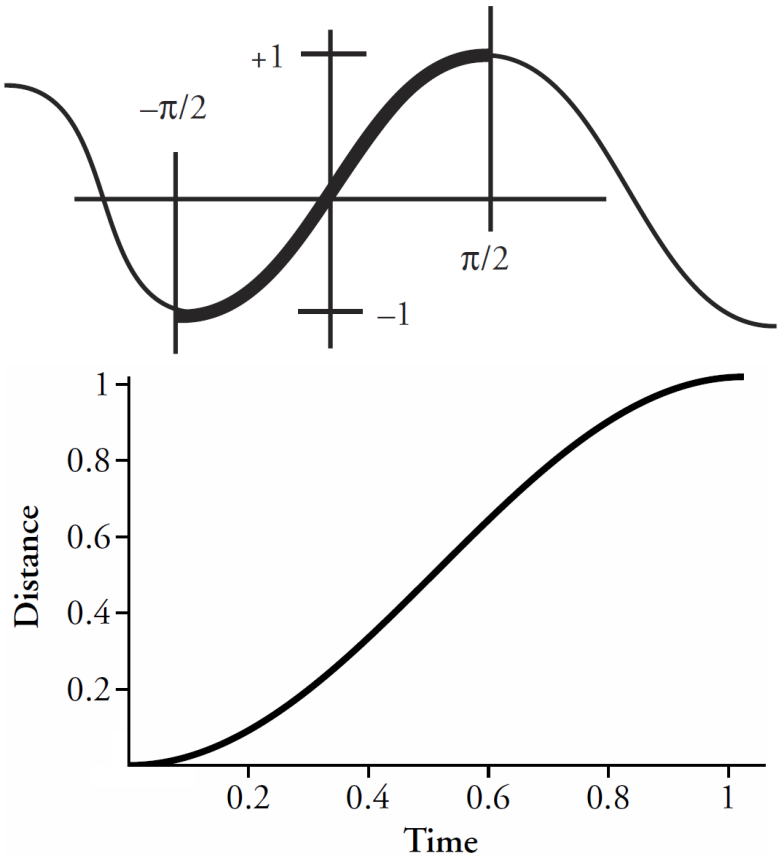
Rozpędzanie i zatrzymywanie

Rozpędzanie i zatrzymywanie to najbardziej użyteczne i najczęściej stosowane sposoby sterowania prędkością wzdłuż krzywej.

Standardowo zakłada się, że ruch zaczyna i kończy z zerową prędkością i nie ma nagłych skoków prędkości. Może istnieć przedział czasowy, w którym prędkość jest stała.

Użycie funkcji sinus

Prostym sposobem określenia rozpędzania i zatrzymywania jest użycie funkcji sinus w przedziale $[-\pi/2, \pi/2]$.



Funkcja odległości *ease* jest wówczas postaci:

$$s = ease(t) = \frac{\sin(t\pi - \frac{\pi}{2}) + 1}{2}.$$

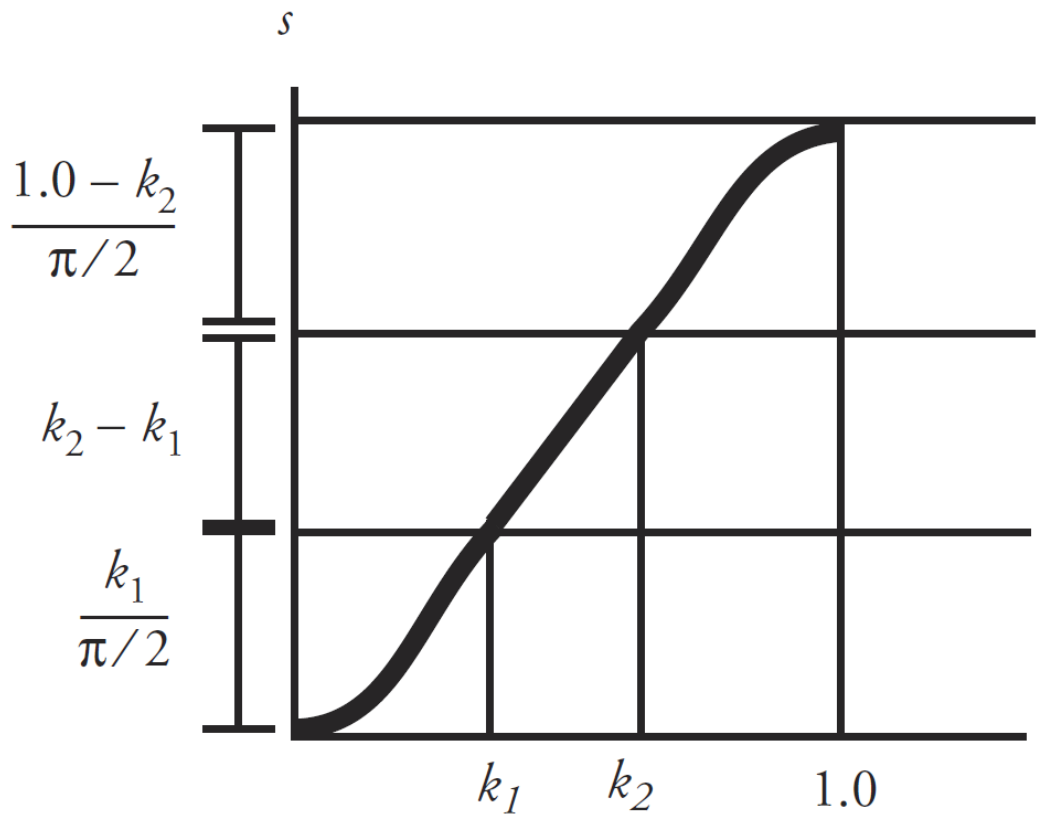
Szybkość zmian zmiennej s nie jest stała w żadnym przedziale czasu, ale zawsze przyspiesza lub zwalnia. Pochodna tej funkcji jest równa 0 dla $t = 0$ i $t = 1$. Oznacza to płynne przyspieszenie od bezruchu w pozycji początkowej i płynne spowolnienie ruchu do zatrzymania w położeniu końcowym.

Przyspieszanie i zwalnianie przy użyciu funkcji kawałkami sinusoidalnej

Jeśli chcemy dostać funkcję odległości zapewniającą ruch ze stałą prędkością w pewnym przedziale czasu, to możemy ją określić w przedziałach za pomocą funkcji sinus, a pomiędzy nimi wprowadzić przedział, w którym funkcja jest liniowa.

Należy zadbać o zgodność stycznych do wykresów fragmentów sinusoid z łączącym je odcinkiem, aby zapewnić ciągłość pochodnej funkcji odległości. Można to zrobić różnymi sposobami.

Załóżmy, że w przedziale $[0, k_1]$ chcemy przyspieszać, w $[k_2, 1]$ zwalniać, a w $[k_1, k_2]$ utrzymywać stałą prędkość.



Wzór na funkcję odległości jest następujący:

$$ease(t) = \begin{cases} \frac{2k_1}{\pi f}\left(\sin\left(\frac{t\pi}{2k_1} - \frac{\pi}{2}\right) + 1\right) & t < k_1, \\ \left(\frac{2k_1}{\pi} + t - k_1\right)/f & k_1 \leqslant t < k_2, \\ \left(\frac{2k_1}{\pi} + k_2 - k_1 + \left(\frac{2(1-k_2)}{\pi}\sin\left(\frac{\pi(t-k_2)}{2(1-k_2)}\right)\right)\right)/f & k_2 \leqslant t, \end{cases}$$

gdzie $f = 2k_1/\pi + k_2 - k_1 + 2(1 - k_2)/\pi$.

Przyspieszanie i zwalnianie za pomocą wielomianu trzeciego stopnia

Do przybliżania sinusoidalnej funkcji odległości może być użyty wielomian:

$$s = -2t^3 + 3t^2.$$

Przyspieszanie i zwalnianie za pomocą wielomianu trzeciego stopnia

Do przybliżania sinusoidalnej funkcji odległości może być użyty wielomian:

$$s = -2t^3 + 3t^2.$$

Wykres tego wielomianu przechodzi przez $(0, 0)$ i $(1, 1)$ przy czym styczne do wykresu w tych punktach są poziome (czyli tangens ich kąta nachylenia jest równy 0). Wadą tej funkcji jest brak przedziału o stałej prędkości.

Stałe przyspieszenie

W celu uniknięcia obliczania funkcji przestępnych przy zachowaniu możliwości otrzymania przedziału o stałej prędkości między rozpędzaniem i zwalnianiem musimy przyjąć pewne założenia dotyczące przyspieszenia.

Stałe przyspieszenie

W celu uniknięcia obliczania funkcji przestępnych przy zachowaniu możliwości otrzymania przedziału o stałej prędkości między rozpędzaniem i zwalnianiem musimy przyjąć pewne założenia dotyczące przyspieszenia.

W najprostszym przypadku (bez rozpędzania i zwalniania) mamy zerowe przyspieszenie i funkcję prędkości o stałej wartości v_0.

Prędkość v_0 zależy od przebytej odległości i spełnia zależność:

$$odleglosc = predkosc \cdot czas.$$

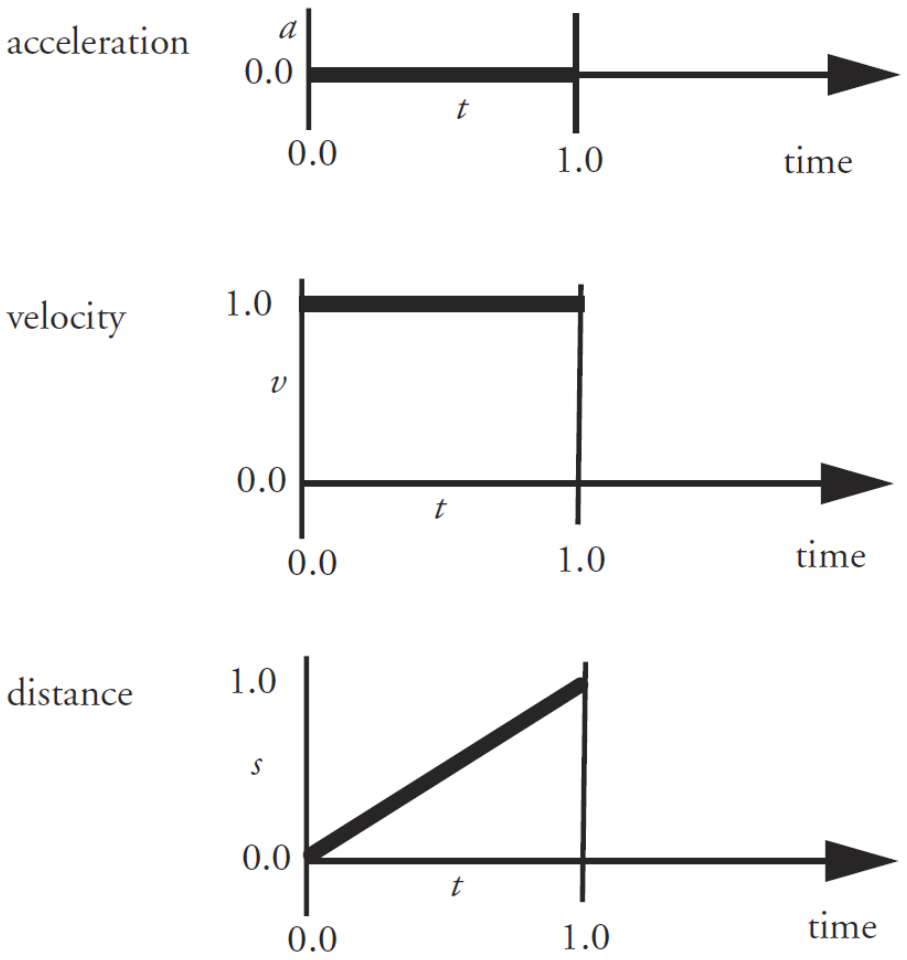
Zatem

$$v_0 = \frac{cala\_odleglosc}{calkowity\_czas}.$$

W przypadku gdy odległość i całkowity czas są znormalizowane mamy $v_0 = 1$.

Funkcja prędkości jest całką z przyspieszenia i wiąże czas z prędkością ruchu wzdłuż krzywej.

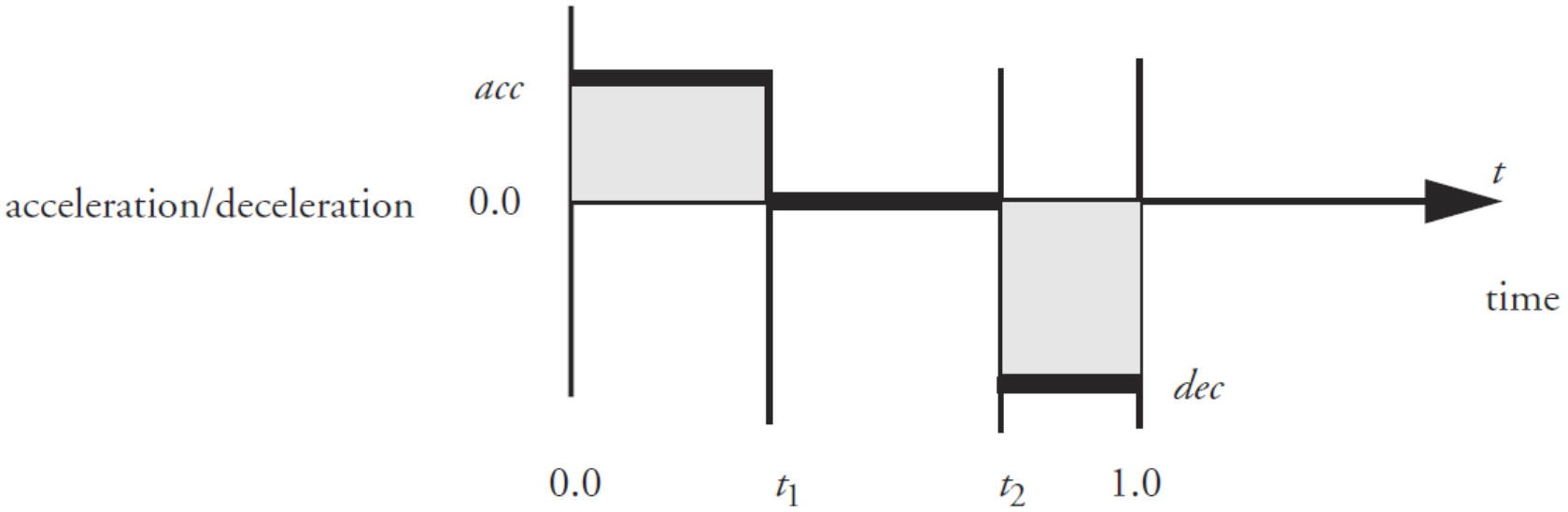
Funkcja opisująca zależność przebytej drogi od czasu jest całką z funkcji opisującej prędkość.



W celu realizacji rozpędzania i zwalniania możemy przyjąć stałe przyspieszenie na początku i pod koniec ruchu oraz zerowe przyspieszenie w środku.

Obiekt ma być nieruchomy w położeniu początkowym i końcowym, więc prędkość początkowa i końcowa są równe 0. Aby to osiągnąć musimy przyjąć, że obszar pod fragmentem wykresu przyspieszenia *aac* ma takie samo pole jak obszar nad fragmentem wykresu *dec*.

Użytkownik może podać trzy z czterech zmiennych (*acc*, *dec*, t_1, t_2), a program policzy wartość czwartej zmiennej spełniającą podany wcześniej warunek.

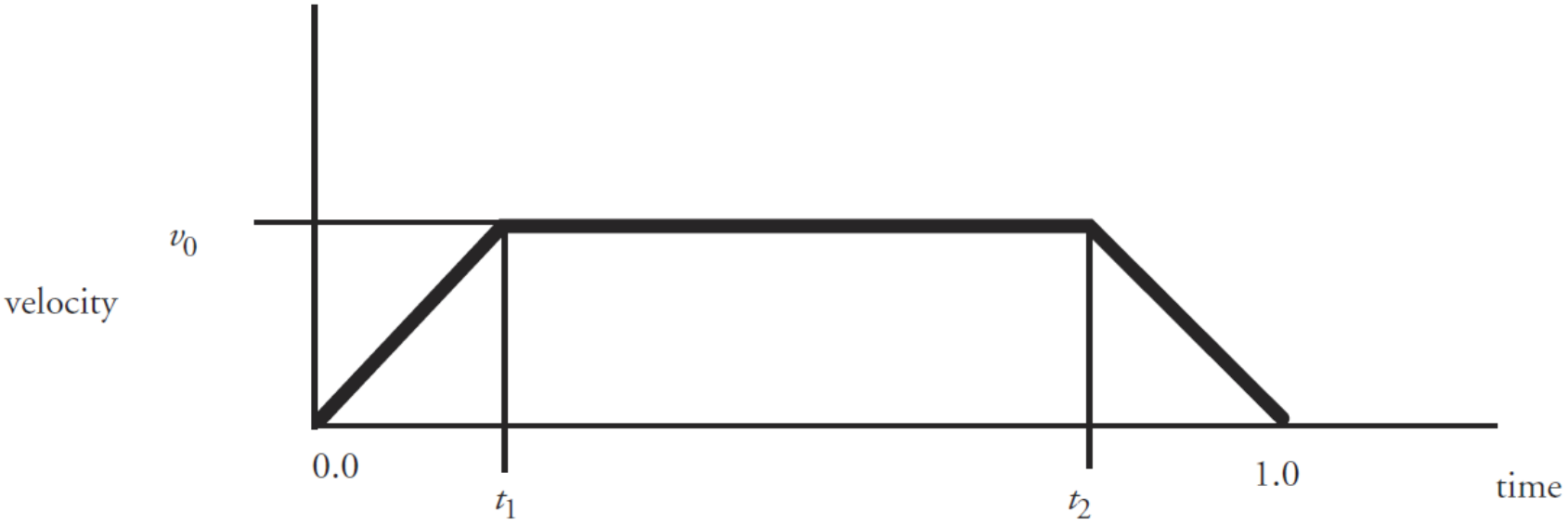


$a = acc \qquad 0 < t < t_1$

$a = 0.0 \qquad t_1 < t < t_2$

$a = dec \qquad t_2 < t < 1.0$

Całkując funkcję opisującą przyspieszenie otrzymujemy funkcję prędkości.



$$v = v_0 \cdot \frac{t}{t_1} \qquad 0.0 < t < t_1$$

$$v = v_0 \qquad t_1 < t < t_2$$

$$v = v_0 \cdot \left(1.0 - \frac{t - t_2}{1.0 - t_2}\right) \qquad t_2 < t < 1.0$$

Przebyta odległość jest polem pod wykresem funkcji prędkości. Jeśli czas i odległość są znormalizowane, to:

$$1 = \frac{1}{2} v_0 t_1 + v_0 (t_2 - t_1) + \frac{1}{2} v_0 (1 - t_2).$$

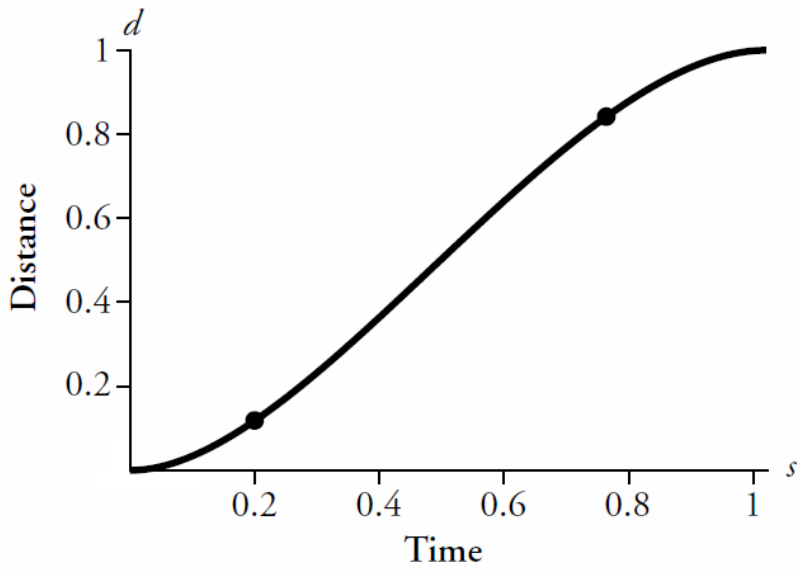
Przebyta odległość jest polem pod wykresem funkcji prędkości. Jeśli czas i odległość są znormalizowane, to:

$$1 = \frac{1}{2} v_0 t_1 + v_0 (t_2 - t_1) + \frac{1}{2} v_0 (1 - t_2).$$

Wykres przyspieszenia nie ma bezpośredniego związku z przebytą odległością, więc bardziej intuicyjne dla użytkownika może być określanie parametrów rozpędzania i zwalniania za pomocą wykresu funkcji prędkości.

W takim przypadku użytkownik podaje wartości dwóch z trzech zmiennych t_1, t_2, v_0, a system obliczy trzecią, aby zapewnić, że przebyta odległość jest równa 1.

Całkując otrzymaną funkcję prędkości otrzymujemy funkcję odległości.



$$d = v_0 \cdot \frac{t^2}{2 \cdot t_1} \qquad 0.0 < t < t_1$$

$$d = v_0 \cdot \frac{t_1}{2} + v_0 \cdot (t - t_1) \qquad t_1 < t < t_2$$

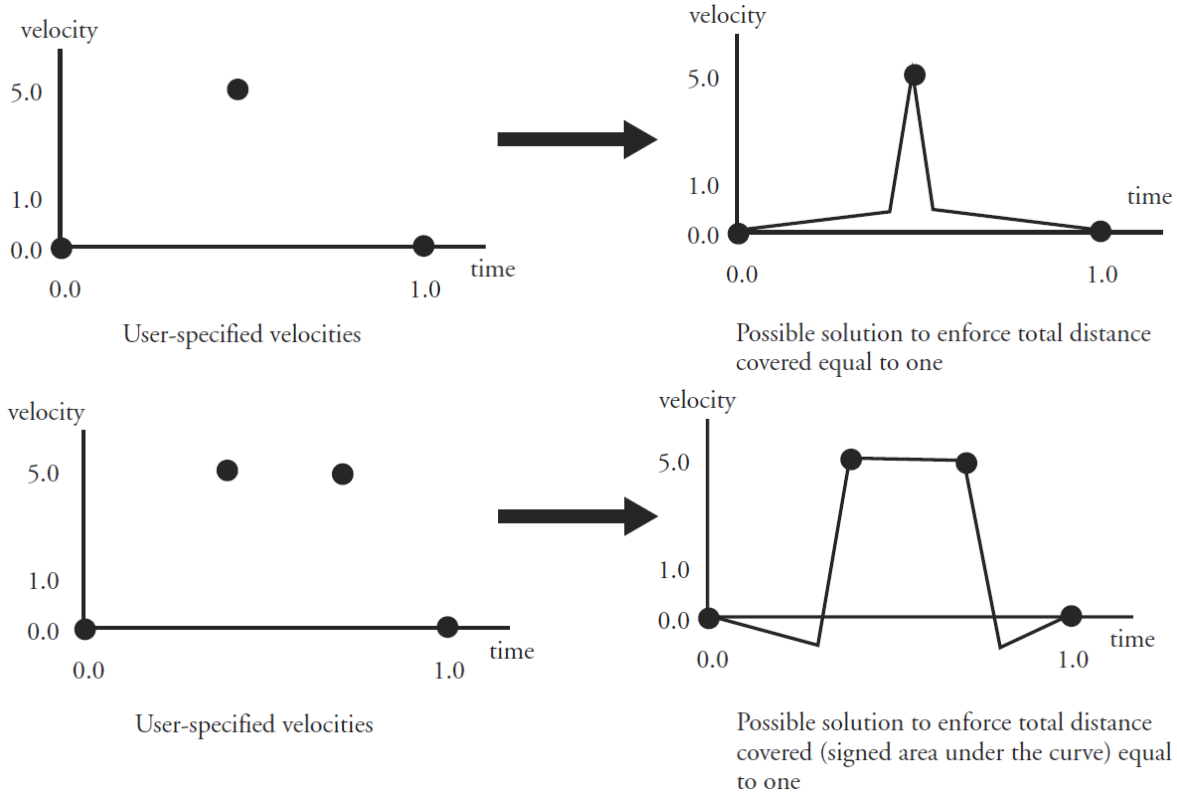
$$d = v_0 \cdot \frac{t_1}{2} + v_0 \cdot (t_2 - t_1) + \left(v_0 - \frac{\left(v_0 \cdot \frac{t - t_2}{1 - t_2} \right)}{2} \right) \cdot (t - t_2) \qquad t_2 < t < 1.0$$

Ogólne funkcje odległości i czasu

Zadanie polegające na przebyciu całej drogi w zadanym przedziale czasu określa średnią prędkość ruchu. Prędkość ta musi być zachowana jeśli użytkownik modyfikuje funkcję opisującą prędkość. Może to być problemem jeśli pracujemy jedynie z wykresem prędkości.

Możliwym rozwiązaniem jest dopuszczenie, aby wykres prędkości „pływał" w górę i dół podczas modyfikowania jego kształtu. W ten sposób średnia prędkość może zostać zachowana.

Innym sposobem sterowania prędkością polega na ustaleniu prędkości w punktach kluczowych i dostosowaniu funkcji prędkości między nimi w celu skorygowania prędkości średniej. To jednak może doprowadzić do niespodziewanych zmian kształtu wykresu funkcji prędkości, np. nienaturalnych „pików”.

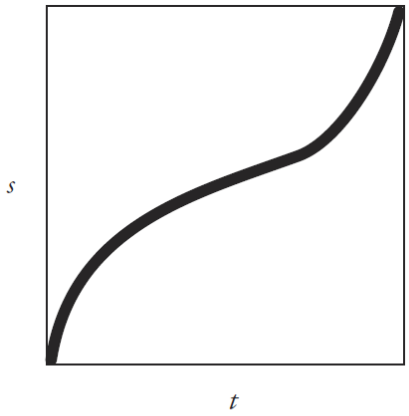


User-specified velocities

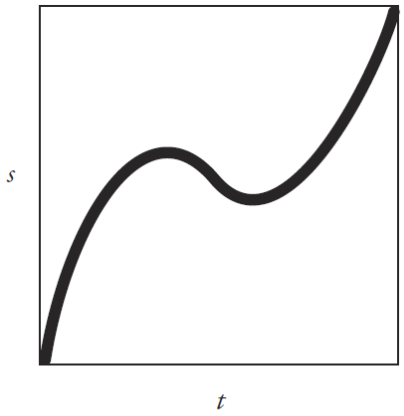
Possible solution to enforce total distance covered equal to one

User-specified velocities

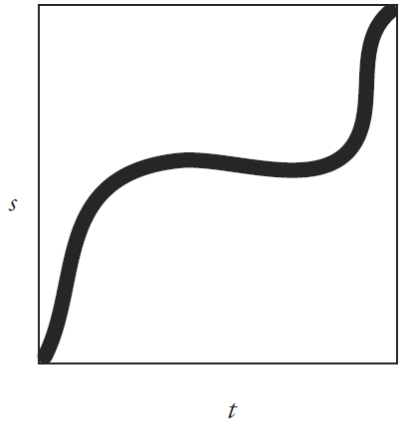
Possible solution to enforce total distance covered (signed area under the curve) equal to one



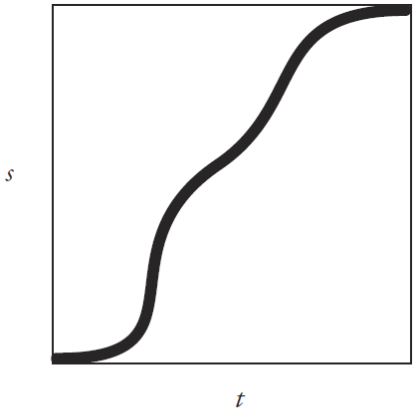
Starts and ends abruptly



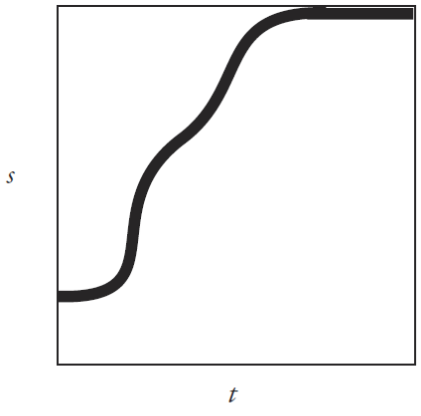
Backs up



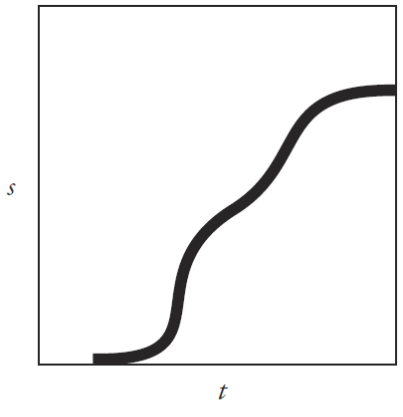
Stalls



Smoothly starts and stops



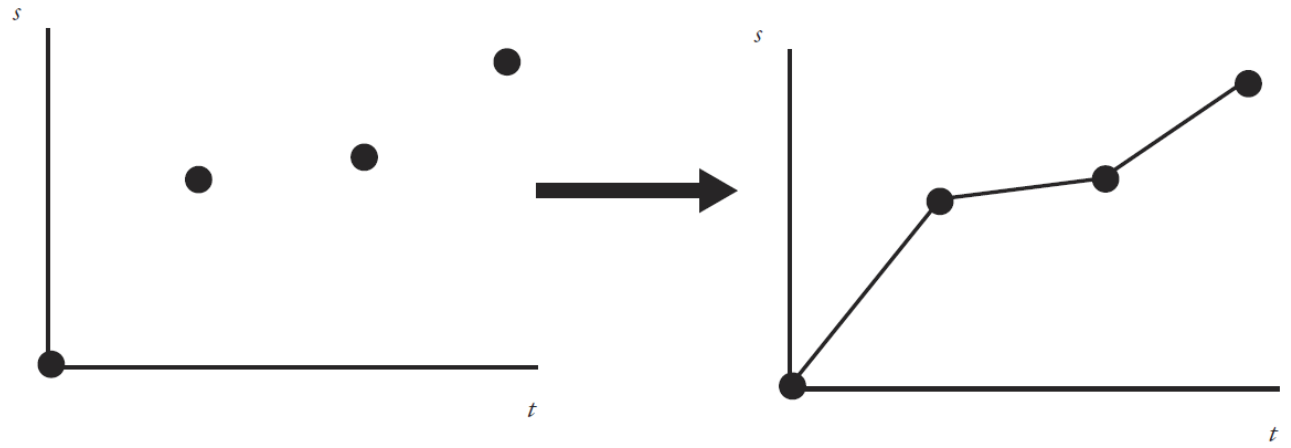
Starts partway along the curve
and gets to the end before $t = 1.0$



Waits a while before starting
and does not reach the end

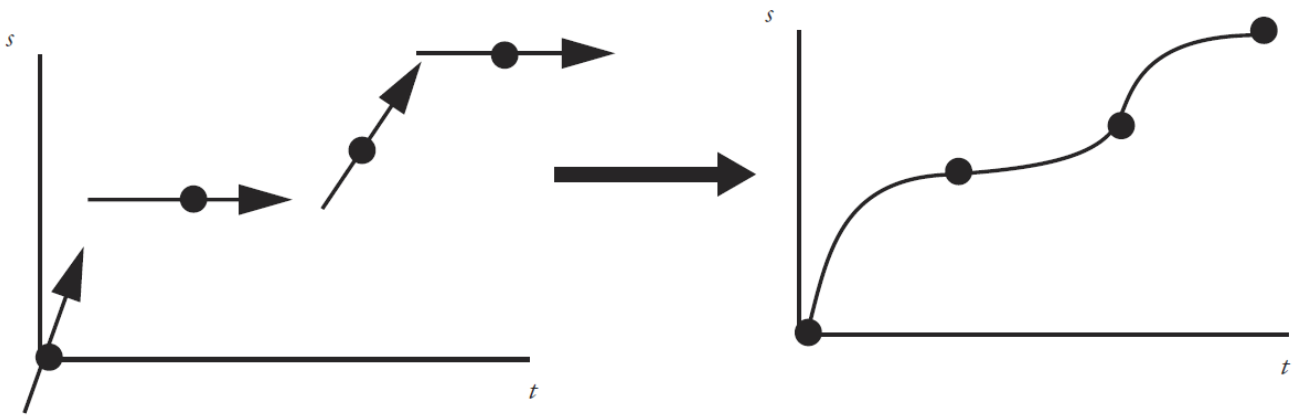
Sterowanie ruchem często wymaga określenia położeń i prędkości w wybranych chwilach, tzn. określamy n-kę $(t_i, s_i, v_i, a_i, \ldots)$, gdzie s_i – położenie, v_i – prędkość, a_i – przyspieszenie, a t_i – chwila, w której te wielkości mają być osiągnięte.

Warunek określony przez pary (t_i, s_i), taki że prędkość, przyspieszenie itd. należy dobrać tak, aby zapewnić osiągnięcie odpowiedniego położenia w chwili t_i nazywamy warunkiem rzędu 0. Warunki rzędu 1 określone są przez (t_i, s_i, v_i).



Distance-time constraints specified

Resulting curve



Velocity-distance-time constraints specified

Resulting curve

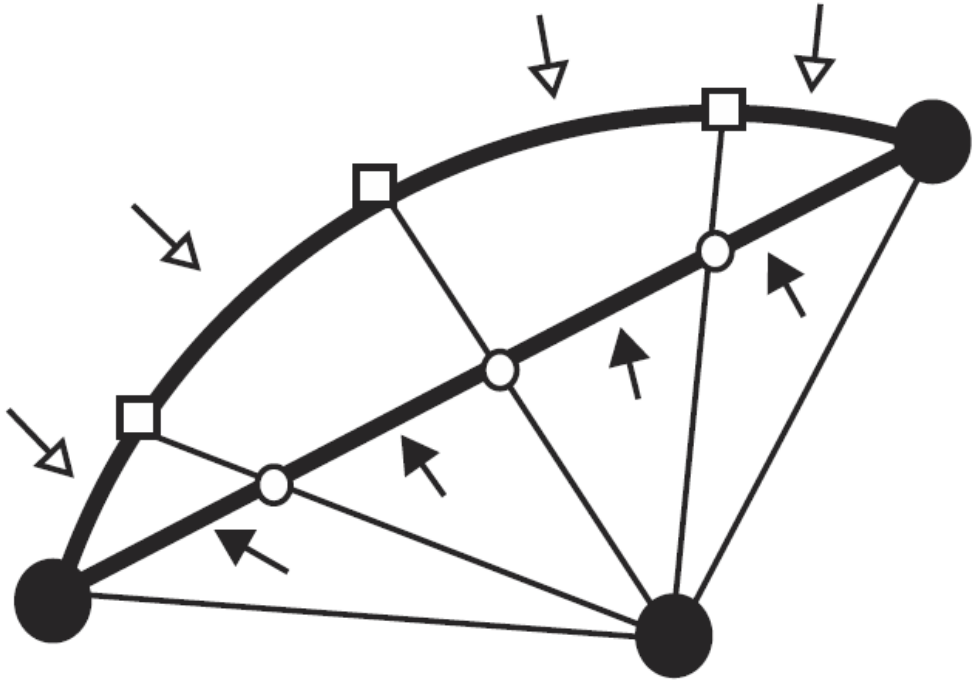
Interpolacja kwaternionów

Kwaterniony są często stosowane do reprezentowania położeń kątowych m.in ze względu na łatwość interpolowania, brak zjawiska blokady przegubu.

Ponieważ długość kwaternionu nie ma wpływu na reprezentowane położenie kątowe zwykle za kanoniczną reprezentację położenia kątowego przyjmuje się kwaternion unormowany. Kwaterniony unormowane można interpretować jak punkty na sferze jednostkowej w przestrzeni czterowymiarowej.

Mając dane dwa położenia kątowe reprezentowane przez kwaterniony unormowane możemy otrzymać położenie pośrednie dokonując liniowej interpolacji każdej z czterech współrzędnych. Otrzymany kwaternion leżący na odcinku, którego końcami są dane dwa kwaterniony.

Jeśli otrzymane punkty są równoodległe, to odpowiadające położenia kątowe nie odpowiadają ruchowi obrotowemu ze stałą prędkością kątową. Jest tak ponieważ odpowiednie kwaterniony unormowane nie dzielą łuku na sferze jednostkowej na równe łuki.



linearly interpolated intermediate points

projection of intermediate points onto circle

equal intervals

unequal intervals

Pośrednie położenia kątowe odpowiadające stałej prędkości obrotowej mogą być otrzymane za pomocą interpolacji dokonanej bezpośrednio na sferze jednostkowej.

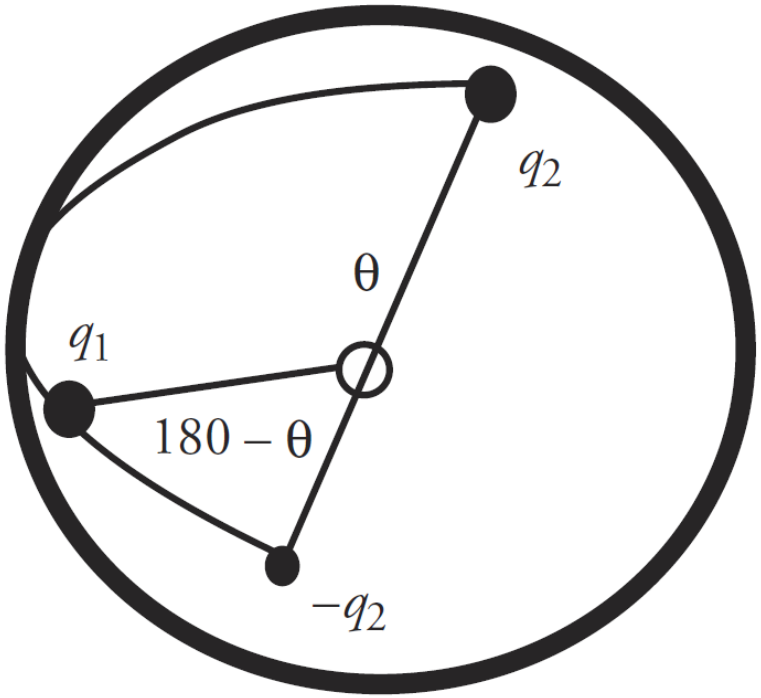
Dowolny kwaternion i jego kwaternion przeciwny reprezentują to samo położenie kątowe. Oznacza to, że interpolacja między położeniami kątowymi danymi przez q_1 i q_2 może być zastąpiona przez interpolację między q_1 i $-q_2$. Różnica polega na tym, że jeden z łuków jest dłuższy niż drugi.

Zwykle bardziej pożądane jest wybranie krótszego łuku, ponieważ reprezentuje on bardziej bezpośrednie przejście między danymi położeniami kątowymi.

Wyboru można dokonać badając kąt między kwaternionami:

$$\cos\theta = \frac{q_1 \cdot q_2}{|q_1||q_2|} = \frac{s_1 s_2 + \mathrm{v}_1 \cdot \mathrm{v}_2}{|q_1||q_2|}.$$

Jeśli $\cos\theta$ jest dodatni, to łuk między q_1 i q_2 jest krótszy. W przeciwnym przypadku krótszy jest łuk między q_1 i $-q_2$.



Interpolacja łukowa (ang. spherical linear interpolation, w skrócie slerp) kwaternionów unormowanych q_1, q_2:

$$slerp(q_1, q_2, u) = \frac{\sin(1-u)\theta}{\sin\theta} q_1 + \frac{\sin u\theta}{\sin\theta} q_2,$$

gdzie $u \in [0, 1]$, a θ jest kątem pomiędzy q_1 i q_2.

Interpolacja łukowa (ang. spherical linear interpolation, w skrócie slerp) kwaternionów unormowanych q_1, q_2:

$$slerp(q_1, q_2, u) = \frac{\sin(1-u)\theta}{\sin\theta} q_1 + \frac{\sin u\theta}{\sin\theta} q_2,$$

gdzie $u \in [0, 1]$, a θ jest kątem pomiędzy q_1 i q_2.

Użycie slerp dla ciągu położeń kątowych ma tę samą wadę, co interpolacja liniowa w przestrzeni euklidesowej tzn. brak ciągłości pochodnej pierwszego rzędu.

Shoemake w 1985 roku zaproponował użycie krzywych interpolacyjnych określonych w sposób przypominający kubiczne krzywe Béziera.

Załóżmy, że mamy dany ciąg punktów na płaszczyźnie $(\ldots, \mathrm{p}_{n-1}, \mathrm{p}_n, \mathrm{p}_{n+1}, \ldots)$ tzw. punkty interpolowane.

Między każde dwa kolejne punkty interpolowane wstawiamy dwa wewnętrzne punkty kontrolne. Otrzymujemy w ten sposób kubiczne krzywe Béziera.

Załóżmy, że mamy dany ciąg punktów na płaszczyźnie $(\ldots, p_{n-1}, p_n, p_{n+1}, \ldots)$ tzw. punkty interpolowane.

Między każde dwa kolejne punkty interpolowane wstawiamy dwa wewnętrzne punkty kontrolne. Otrzymujemy w ten sposób kubiczne krzywe Béziera.

Z punktem p_n będą związane dwa punkty kontrolne, z których jeden $a_n = \frac{1}{2}(p_n + (p_n - p_{n-1}) + p_{n+1})$ wstawiamy za p_n, a drugi $b_n = p_n + (p_n - a_n)$ wstawiamy przed nim.

Kubiczne krzywe Béziera dane przez $p_{n-1}, a_{n-1}, b_n, p_n$ oraz $p_n, a_n, b_{n+1}, p_{n+1}$ łączą się w punkcie p_n z ciągłością pochodnych pierwszego rzędu.

Pozostaje kwestia punktów kontrolnych dla pierwszej i ostatniej pary punktów interpolowanych. Konstrukcja jest podobna, np. $a_0 = p_1 + (p_1 - p_2)$.

Pozostaje kwestia punktów kontrolnych dla pierwszej i ostatniej pary punktów interpolowanych. Konstrukcja jest podobna, np. $a_0 = p_1 + (p_1 - p_2)$.

Przedstawioną konstrukcję łatwo jest dostosować do obliczeń na kwaternionach. Zamiast dodawania wektorów, należy składać obroty używając mnożenia kwaternionów. Wyznaczanie środka odcinka zastępujemy przez interpolację łukową zaimplementowaną przez dodawanie kwaternionów i normalizację sumy.

Po obliczeniu punktów kontrolnych stosujemy algorytm de Casteljau w celu wyznaczenia punktów leżących na krzywej.

Załóżmy, że chcemy otrzymać kwaternion p reprezentujący położenie kątowe w $\frac{1}{3}$ ruchu obrotowego między położeniami reprezentowanymi przez q_1 i q_2.

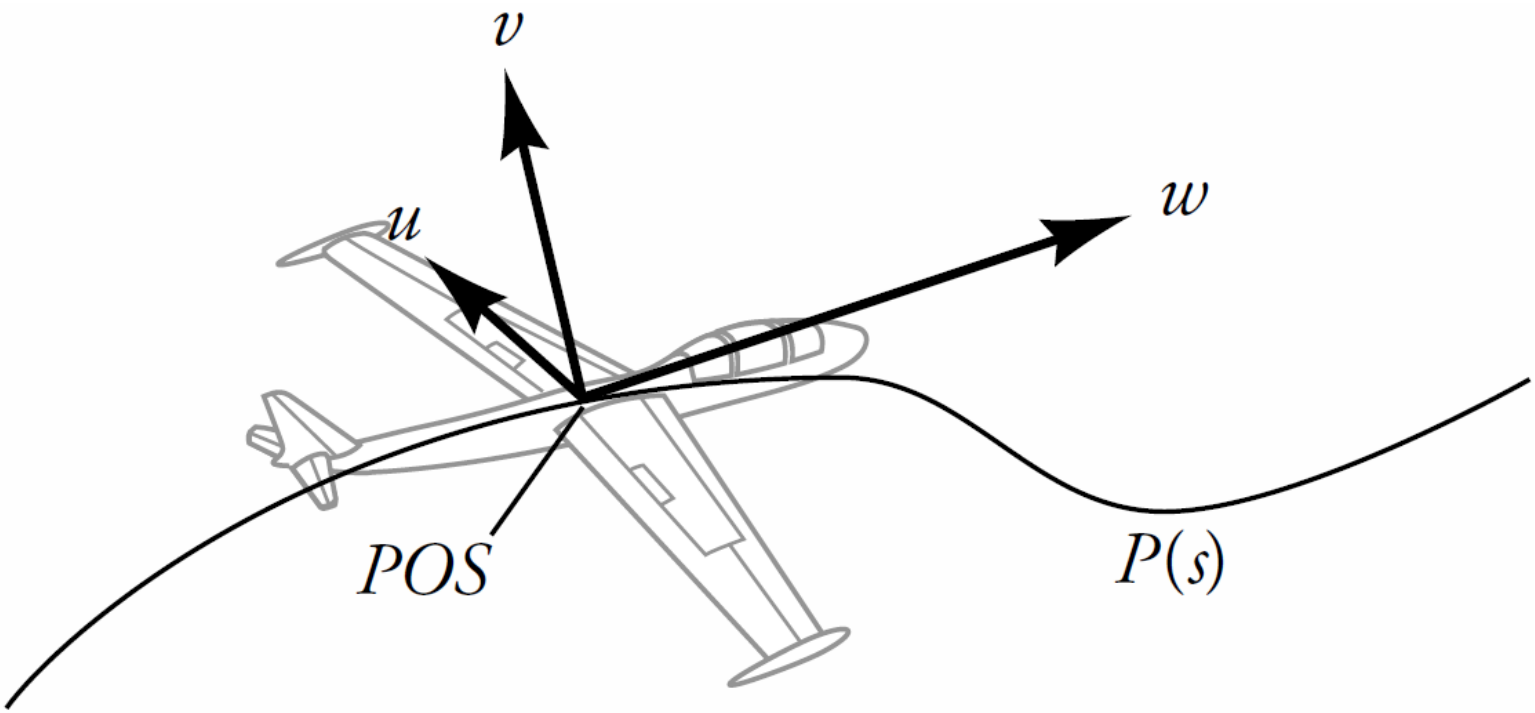
$$
\begin{aligned}
p_1 &= slerp(q_n, a_n, 1/3) \\
p_2 &= slerp(a_n, b_{n+1}, 1/3) \\
p_3 &= slerp(b_{n+1}, q_{n+1}, 1/3) \\
p_{12} &= slerp(p_1, p_2, 1/3) \\
p_{23} &= slerp(p_2, p_3, 1/3) \\
p &= slerp(p_{12}, p_{23}, 1/3)
\end{aligned}
$$

Zorientowanie wzdłuż ścieżki

Zwykle dla animowanego obiektu określa się lokalny układ współrzędnych (u, v, w).

Załóżmy, że mamy prawoskrętny układ współrzędnych a jego początek leży w punkcie położonym na ścieżce P (wyznaczony jest na podstawie parametryzacji łukowej). Oznaczmy ten punkt przez POS.

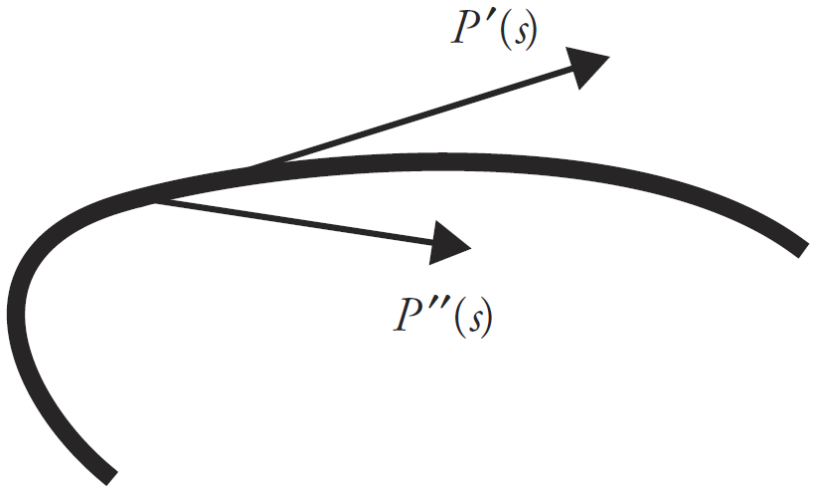
Kierunek, w którym obiekt jest skierowany utożsamiamy z kierunkiem osi w, wektor „do góry" obiektu jest wersorem osi v, a oś u lokalnego układu jest prostopadła do pozostałych dwóch osi.


v
u
w
POS
P(s)

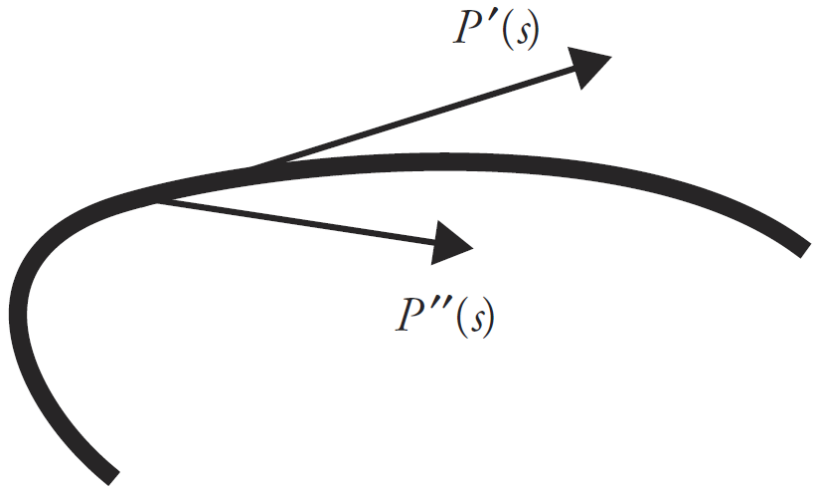
Jeśli obiekt lub kamera porusza się wzdłuż krzywej, to jego położenie kątowe może być określone przez atrybuty geometryczne tej krzywej. Układ Freneta jest to układ współrzędnych (u, v, w) określony przez wektor styczny i normalny krzywej w ustalonym punkcie i obracający się podczas ruchu wzdłuż krzywej.

$$\mathrm{w} = \frac{\mathrm{P}'(s)}{\|\mathrm{P}'(s)\|},$$
$$\mathrm{u} = \frac{\mathrm{P}''(s) \times \mathrm{P}'(s)}{\|\mathrm{P}''(s) \times \mathrm{P}'(s)\|},$$
$$\mathrm{v} = \mathrm{u} \times \mathrm{w}.$$

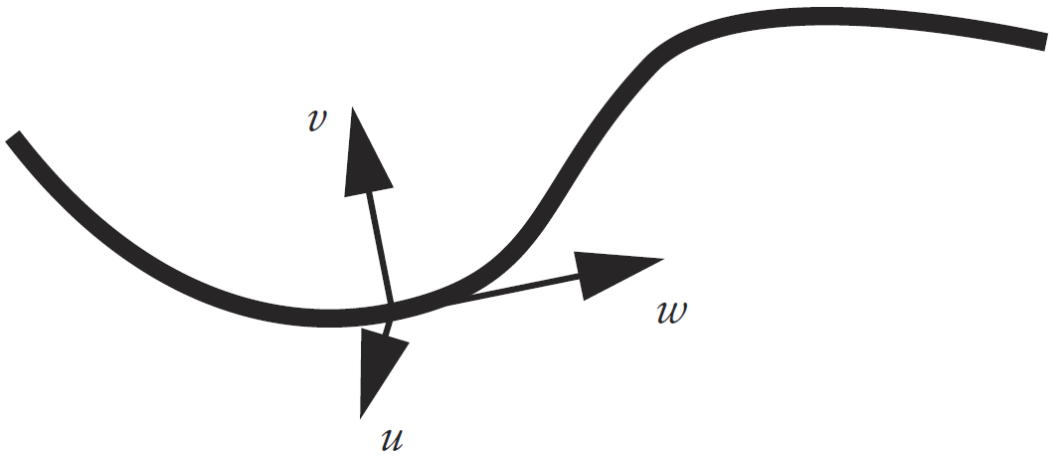
Wektory pochodnych krzywej



Wektory pochodnych krzywej

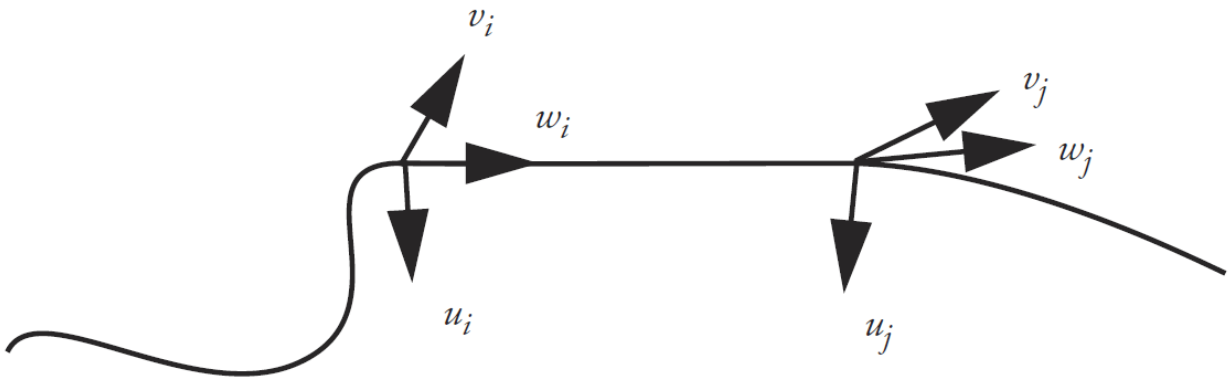


Układ Freneta



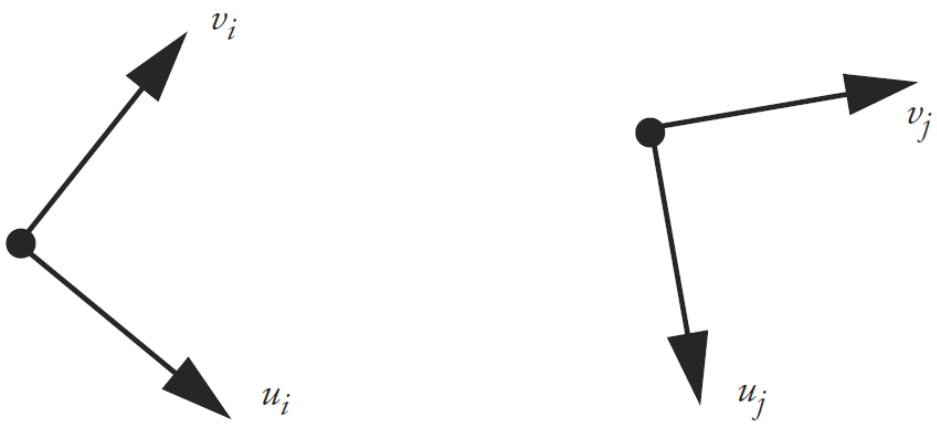
Układ Freneta stwarza pewne problemy przy bezpośrednim stosowaniu do określania położeń kątowych kamery lub obiektu w jego ruchu wzdłuż krzywej:

- nie ma naturalnego pojęcia „kierunku do góry” – zwrot wektora v jest związany tylko ze zwrotem pochodnej drugiego rzędu,
- układ jest nieokreślony jeśli krzywizna krzywej jest zerowa, np. gdy $\mathrm{P}''(s) = 0$. Z tym problemem można sobie poradzić poprzez interpolację układów Freneta na końcach fragmentów o zerowej krzywiźnie (odcinka).



Układy Freneta na końcach przyległych do odcinka krzywych mają wspólny kierunek osi w. Zatem pozostałe osie wystarczy obrócić wokół osi w.

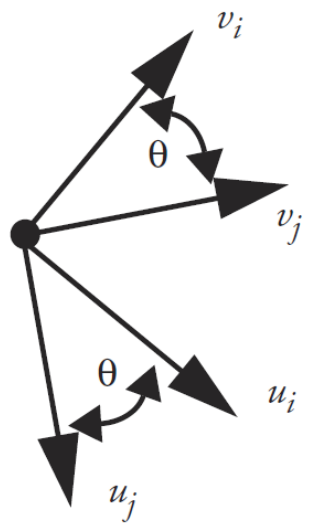
Widok wzdłuż wspólnego wektora w



Ponieważ wersory osi są unormowane możemy obliczyć kosinus kąta obrotu dla całego odcinka obliczając iloczyn skalarny:

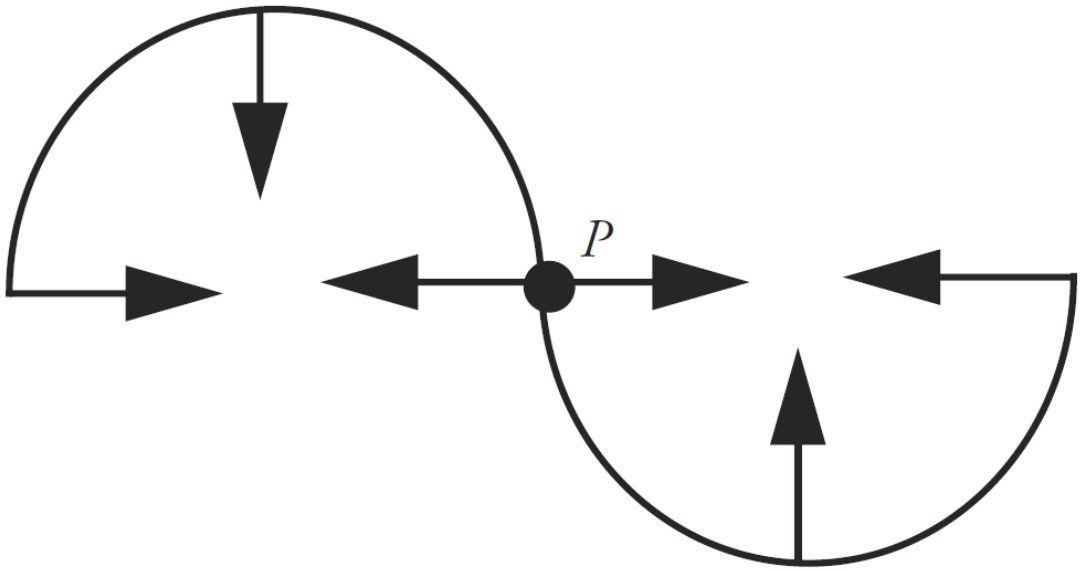
$$\theta = \arccos(\mathsf{v}_1 \cdot \mathsf{v}_2).$$

Następnie możemy dokonać interpolacji liniowej tego obrotu wzdłuż odcinka.



W przypadku gdy wektor normalny ma nieciągłość problem jest jeszcze trudniejszy.

Rozważmy dwa półokręgi:



- Ruch otrzymany z użyciem układu Freneta jest często przesadny i nienaturalny. Często wektor styczny nie określa kierunku, do którego podąża poruszający się obiekt, choć jest kierunkiem ruchu w sensie analitycznym.
- Utożsamianie wektora v z kierunkiem do góry może powodować, że obiekt będzie się gwałtownie obracał wokół stycznej do toru nawet jeśli ten tor łagodnie wije się po otaczającej scenie.

Przy modelowaniu pochylania się podczas wejścia w zakręt wektor normalny (osi u) wskazuje, w którą stronę należy skręcić i może być użyty do określenia wielkości nachylenia. Na przykład jego składowa pozioma (rzut na płaszczyznę xz) może być użyta do wskazania kierunku i kąta nachylenia.

Inny efekt animator może uzyskać obracając obiekt przeciwnie do wektora normalnego, co stwarza wrażenie, że na obiekt działa siła odśrodkowa, jakiej doświadcza osoba jadąca kolejką górską.

Ruch kamery po ścieżce

Najprostszym sposobem określenia położenia kątowego kamery jest wybranie jej środka zainteresowania (COI) w ustalonym punkcie sceny. W ten sposób środek jest jawnie określony i dostępny dla obliczeń wektora kierunku patrzenia, $w = COI - POS$.

Jest to dobra metoda jeśli tor, po którym kamera się porusza okrąża miejsce akcji, na której należy skupić uwagę.

W takim przypadku pozostaje jeszcze jeden stopień swobody w określeniu układu współrzędnych kamery – wektor „do góry". Załóżmy, że wektor „do góry" v ma być zgodny ze zwrotem wersora osi y.

$$\mathrm{w} = \mathit{COI} - \mathit{POS},$$
$$\mathrm{u} = \mathrm{w} \times \mathrm{y},$$
$$\mathrm{v} = \mathrm{u} \times \mathrm{v}.$$

Po obliczeniu u, v, w normujemy je.

$$\mathrm{w} = \mathit{COI} - \mathit{POS},$$
$$\mathrm{u} = \mathrm{w} \times \mathrm{y},$$
$$\mathrm{v} = \mathrm{u} \times \mathrm{v}.$$

Po obliczeniu u, v, w normujemy je.

Jest to dobre rozwiązanie jeśli kamera ma za bardzo nie zbliżać się do obiektu. Przechodzenie blisko środka zainteresowania prowadzi do gwałtownych zmian kierunku patrzenia.

Najprostsza metoda określania wektora kierunku patrzenia polega na użyciu zadanego przyrostu parametru określającego środek zainteresowania.

Jeśli kamera w danej chwili znajduje się w punkcie $\mathrm{P}(s)$, to środkiem zainteresowania będzie punkt $\mathrm{P}(s + \Delta s)$. Do obliczenia tego punktu należy użyć parametryzacji łukowej.

W pobliżu końca krzywej, tj. gdy $s + \Delta s$ jest poza dziedziną krzywej, kierunek patrzenia może być określony przez interpolację między kierunkiem w ostatnim punkcie, w którym został on skonstruowany w opisany sposób i kierunkiem wektora stycznego do krzywej w jej punkcie końcowym.

Często przyjmowanie *COI* w określonym punkcie krzywej prowadzi do otrzymania efektu chwiejącej się kamery.

W takich przypadkach może pomóc przyjęcie *COI* w punkcie otrzymanym przez uśrednienie kilku punktów krzywej.

Jeśli liczba punktów będzie zbyt mała lub będą leżeć blisko siebie, to efekt chwiania może pozostać. Jeśli liczba będzie zbyt duża lub będą one leżały zbyt daleko od siebie, to kierunek patrzenia poruszającej się kamery może zmieniać się niedostatecznie szybko co może wyglądać zbyt statycznie.

Alternatywny sposób określania *COI* polega na wprowadzeniu pewnej funkcji określonej przez tor ruchu kamery. Wówczas położeniu kamery $\mathrm{P}(s)$ odpowiada położenie środka zainteresowania $\mathrm{C}(s)$.

Alternatywny sposób określania *COI* polega na wprowadzeniu pewnej funkcji określonej przez tor ruchu kamery. Wówczas położeniu kamery $\mathrm{P}(s)$ odpowiada położenie środka zainteresowania $\mathrm{C}(s)$.

Podobnie możemy wprowadzić ścieżkę U w celu określenia wektora do góry przy użyciu wektora $\mathrm{U}(s) - \mathrm{P}(s)$.

$$\begin{aligned} \mathrm{w} &= \mathrm{C}(s) - \mathrm{P}(s), \\ \mathrm{u} &= \mathrm{w} \times (\mathrm{U}(s) - \mathrm{P}(s)), \\ \mathrm{v} &= \mathrm{u} \times \mathrm{w}. \end{aligned}$$

Zamiast używać oddzielnej ścieżki do określenia *COI* prosty, ale efektywny sposób polega na ustaleniu jego położenia dla pewnego przedziału czasowego, a potem przeniesieniu go wzdłuż odcinka (z wykorzystaniem funkcji realizującej rozpędzanie i hamowanie), ustaleniu nowego położenia dla pewnej liczby kolejnych klatek itd.