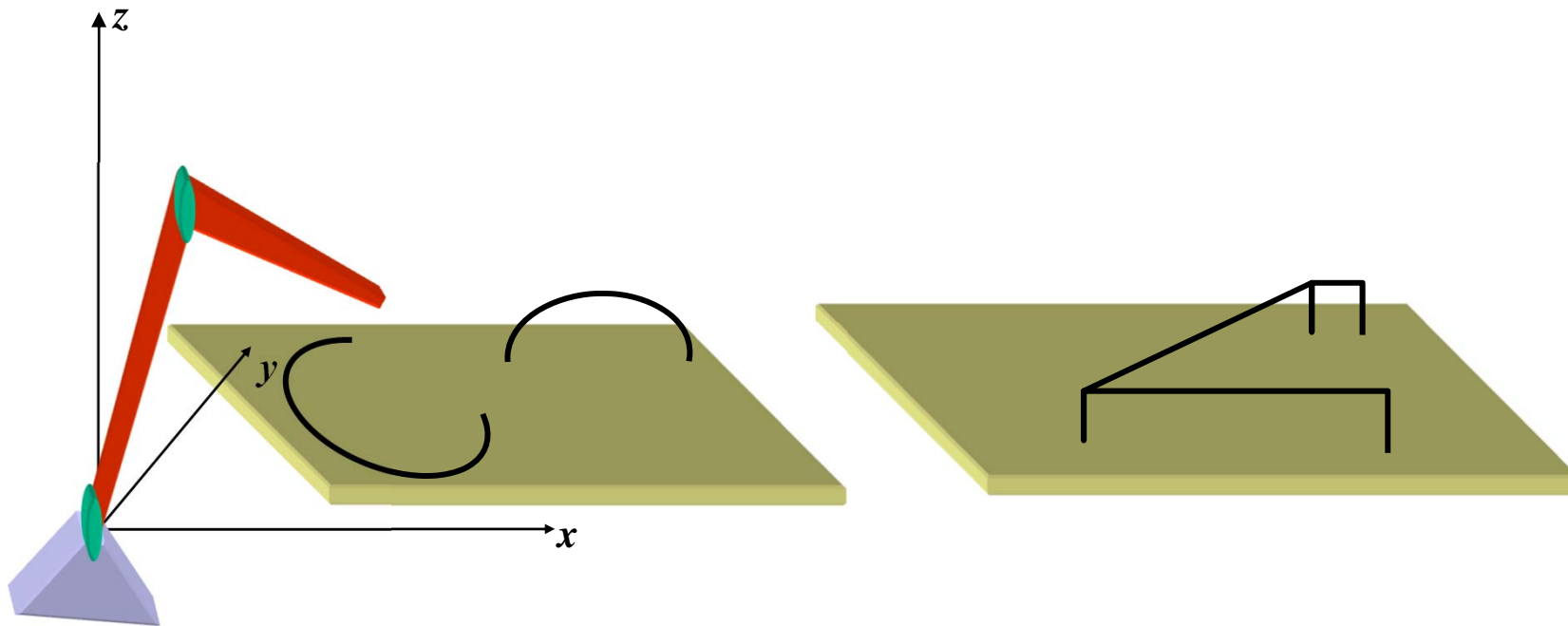
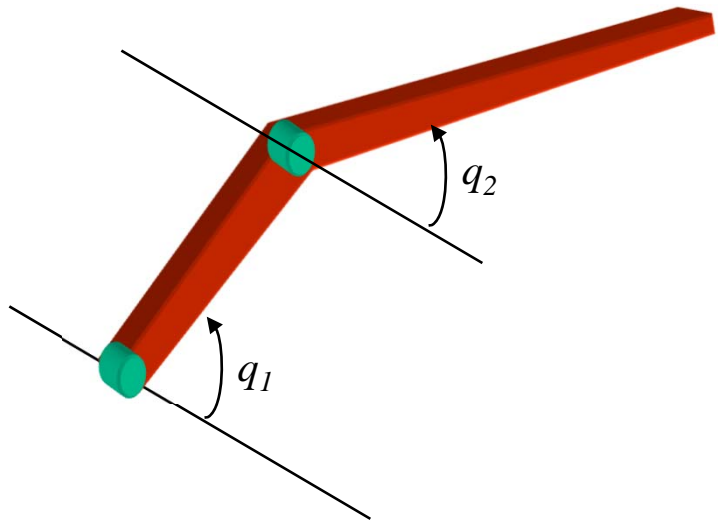


deuxième partie

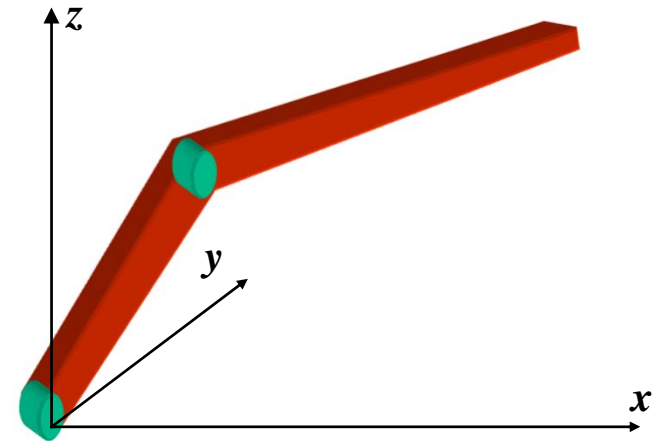
Génération de trajectoires



Les deux espaces du robot

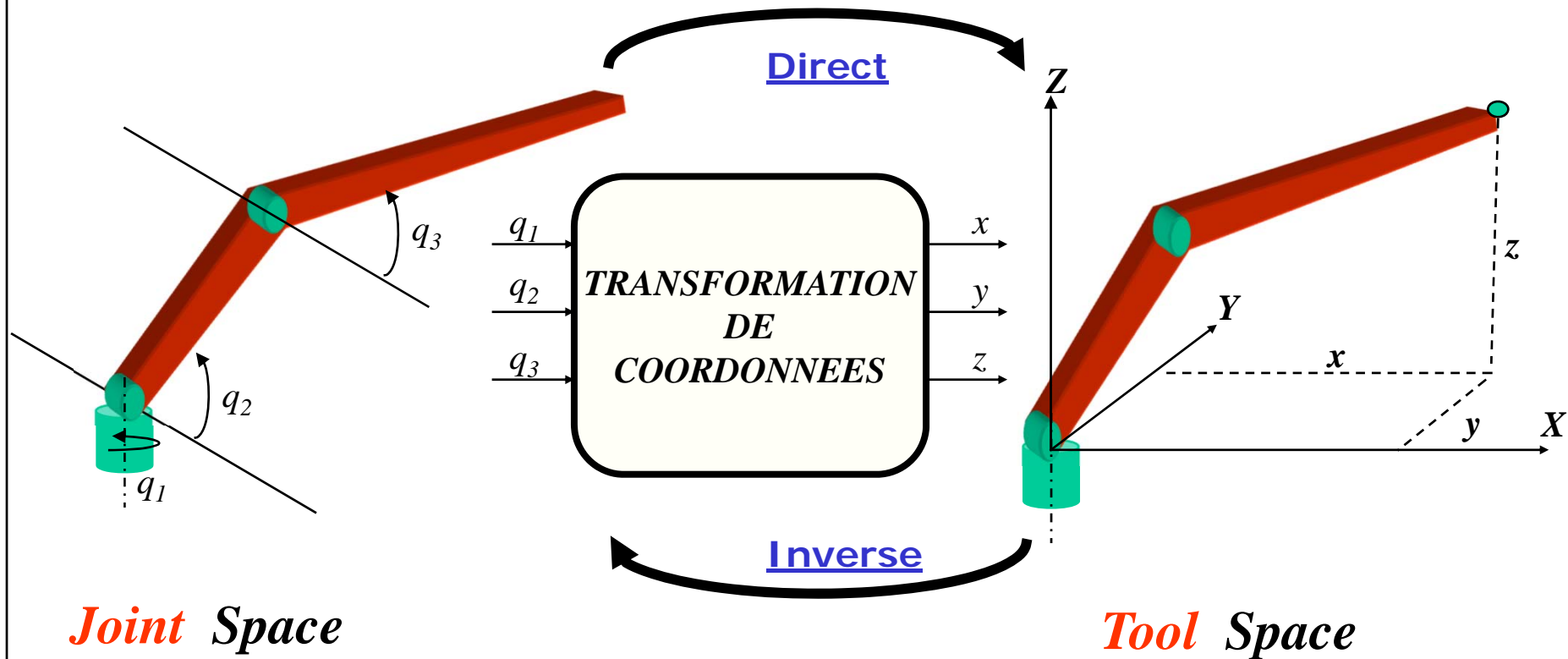


L'espace articulaire



L'espace opérationnel

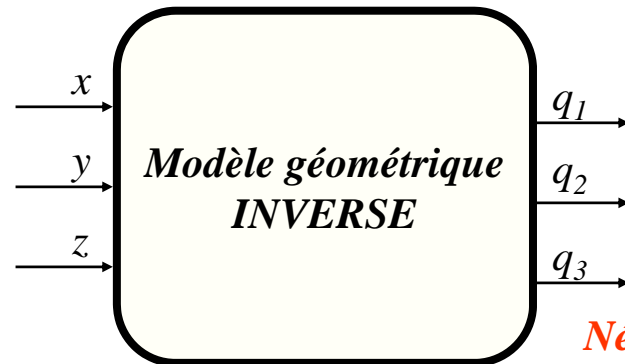
Coordinate Transformation



Le modèle géométrique



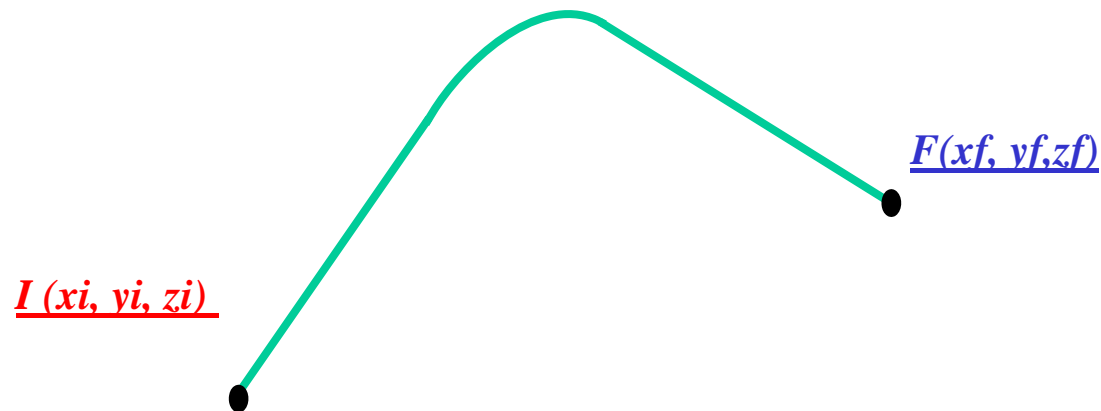
Nécessaire pour l'apprentissage des points!



Nécessaire pour la génération de trajectoires!

Objectif :

Comment aller d'un point $I(x_i, y_i, z_i)$ à un point $F(x_f, y_f, z_f)$



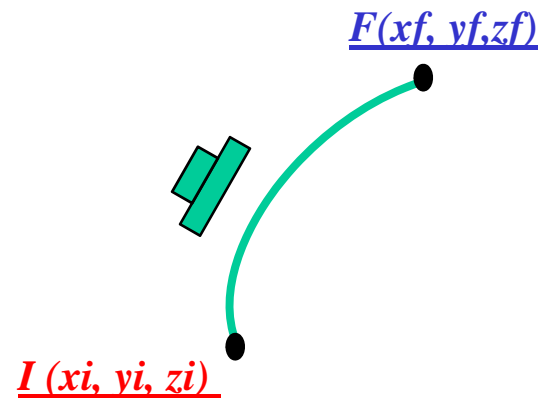
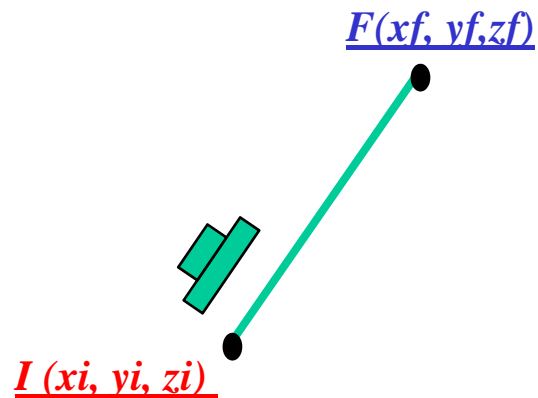
Les deux aspects d'une trajectoire

Aspect spatiale
ou géométrique

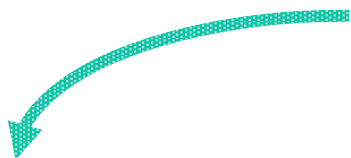
aspect temporel

Ces deux aspects sont ils indépendants?

Non!



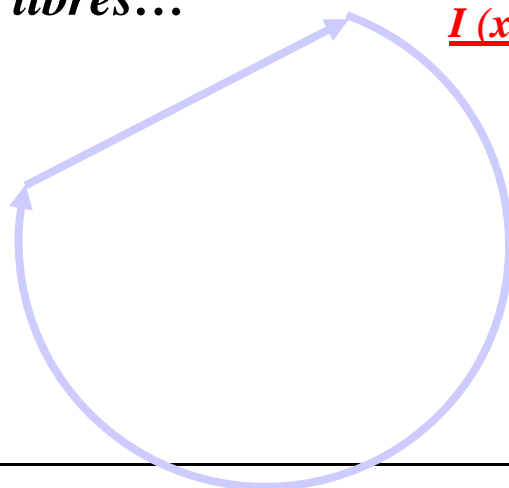
Les deux aspects d'une trajectoire



Géométrie,

Exemples:

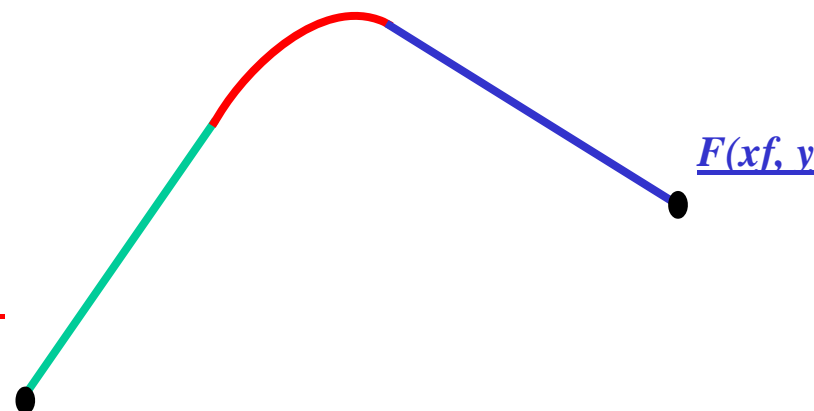
- *Ligne,*
- *Cercle,*
- *Ellipse*
- *Hélice,*
- *Spirale,*
- *Formes libres...*



Aspect temporel : Dynamique

Position en fonction du temps

$I(x_i, y_i, z_i)$



Trajectoires admissibles

Une trajectoire est admissible si:

- elle est réalisable sur le plan géométrique (en aucun cas elle ne sort de l'espace de travail),
- elle est réalisable sur le plan temporel (les vitesses et accélérations sont réalisables par les moteurs)

Pour cela il faut que:

- l'électronique de puissance suive!
- les performances du contrôleur suivent (algorithme de contrôle, puissance de calcul et rapidité d'acquisition)!

Qu'est ce que l'interpolation?



Dans le cas de la génération de trajectoire, nous parlons d'axes interpolés quand leurs mouvements sont géométriquement dépendants.



Les interpolations de base connues dans le monde de la commande des machines sont:

- ✓ l'interpolation linéaire,
- ✓ l'interpolation circulaire (sens horaire et sens anti-horaire).



D'autres interpolations existent:

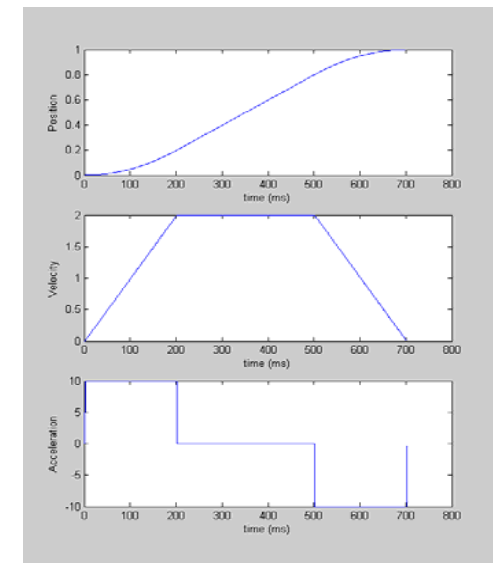
- ✓ l'interpolation spirale,
- ✓ l'interpolation hélicoïdale,
- ✓ l'interpolation spline,
- ✓ ...

Qu'est ce qu'un profil?

↪ Un profil définit la courbe en fonction du temps de la position, vitesse ou accélération.

↪ On parle généralement de profil de vitesse et de profil d'accélération.

(le profil de position se déduit automatiquement)



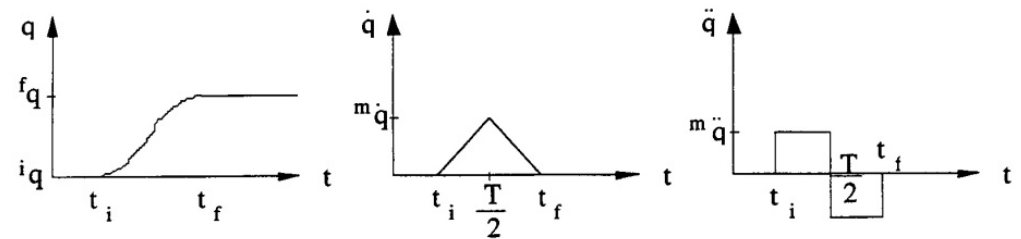
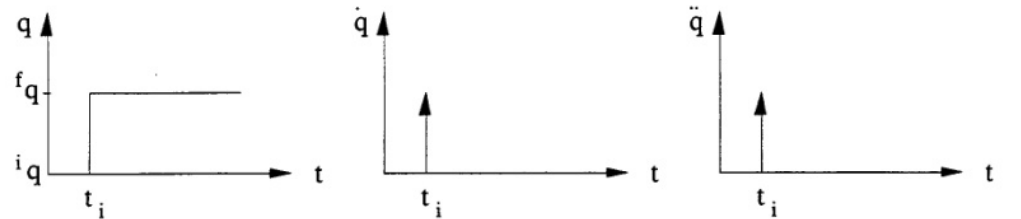
Profils temporels

Interpolation d'ordre 0

loi à accélération bang-bang

loi à accélération trapézoïdale

Le profil le plus connu est le profil trapézoïdal en vitesse (appelé aussi Bang-Bang en accélération et parabolique en position):

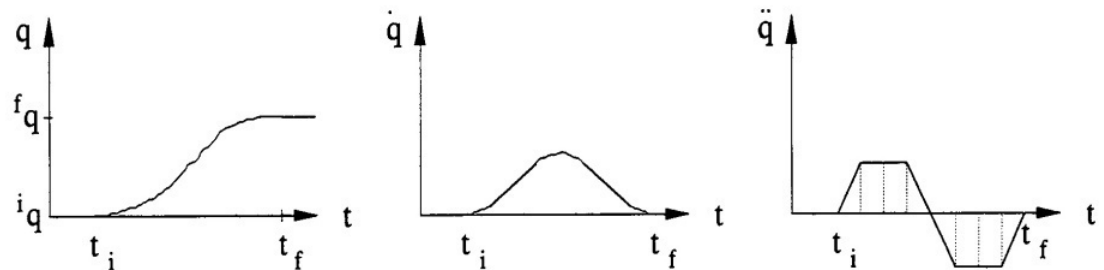


$$Q = f_q - i_q$$

$$T = t_f - t_i$$

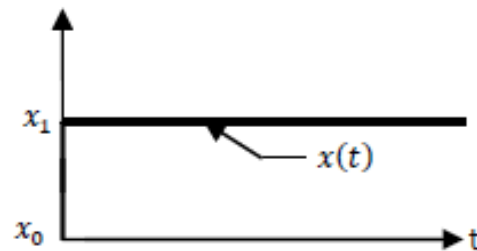
$$m\dot{q} = \frac{2Q}{T}$$

$$m\ddot{q} = \frac{4Q}{T^2}$$



1. l'échelon :

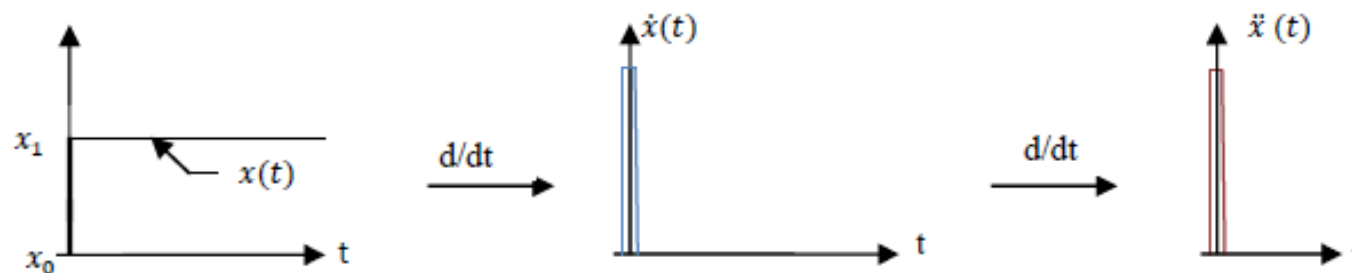
En automatique, on commence souvent par apprendre que la manière la plus simple de réaliser cela consiste à utiliser un échelon (soit un step).



$x(t)$ est alors défini comme suit :

$$x(t) = \begin{cases} x_0 & t \leq 0 \\ x_1 & t > 0 \end{cases}$$

Observons les formes de la vitesse et de l'accélération correspondantes à cette consigne.



Dans le cas de la commande d'actionneurs en générale et particulièrement pour la commande de robots, la trajectoire de type ECHELON ne doit jamais être utilisée.

Remarque :

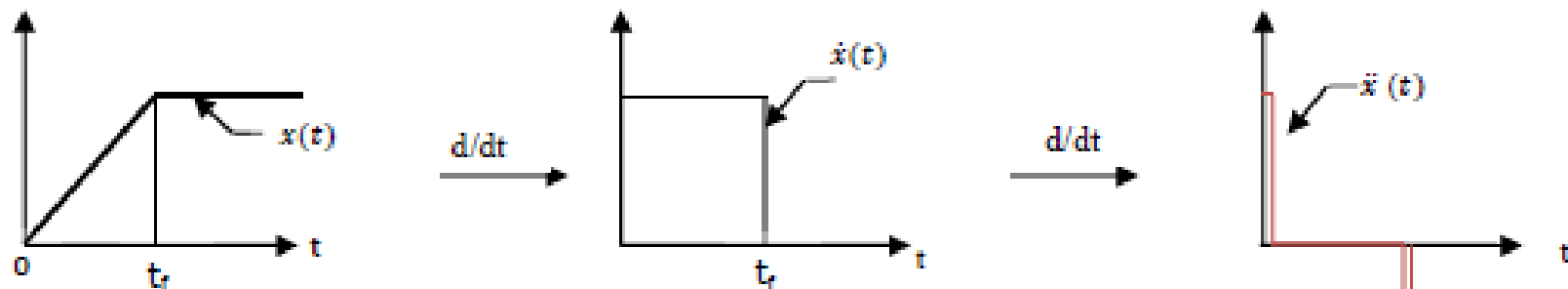
En pratique, la vitesse et accélération (pointe de Dirac) ne sont pas infinies car les dérivées des consignes (de la position, respectivement celle de vitesse) s'effectuent à la cadence de la période d'échantillonnage du contrôle (sampling period T_s).

Les valeurs maximales de la vitesse et de l'accélération sont données par :

$$V_{max} = \frac{\Delta x}{T_s} \text{ et } ACC_{max} = \frac{V_{max}}{T_s} = \frac{\Delta x}{T_s^2}$$

2. La rampe (slope) :

Une autre trajectoire de consigne simple à mettre en oeuvre est la rampe de position. C'est une trajectoire réalisée à vitesse constante tout le long du trajet.



Dans ce cas, nous maîtrisons le temps de réalisation du parcours, la vitesse est ainsi limitée (constante). Le saut d'accélération reste néanmoins toujours brutale mais moins que dans le cas d'un échelon.

Calcul de la trajectoire :

Hypothèses de calcul du trajet:

- t_f (temps de réalisation du parcours) ou V_{max} (Vitesse constante le long du parcours).
- La longueur du trajet ¹ $\Delta x = x_1 - x_0$;

Si V_{max} est imposé alors $t_f = \frac{\Delta x}{V_{max}}$

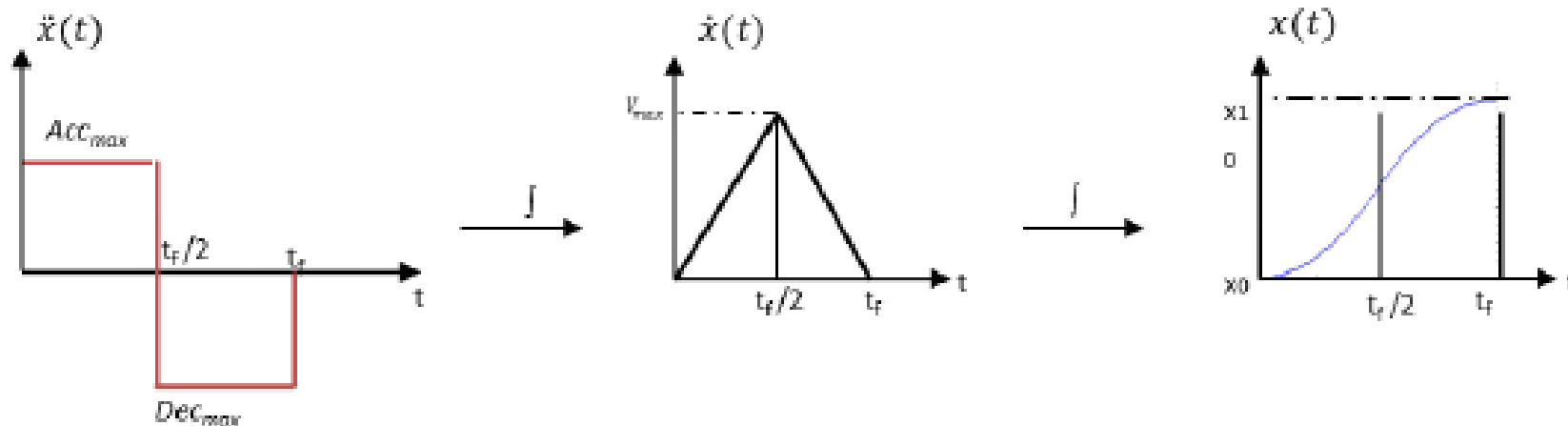
Si t_f est imposé alors $V_{max} = \frac{\Delta x}{t_f}$

Expression de la trajectoire

$$\begin{cases} x_0 & \text{si } t \leq 0 \\ x_0 + V_{max} * t & 0 < t \leq t_f \\ x_1 & t > t_f \end{cases}$$

Le profil triangulaire en vitesse :

Pour éviter les Diracs d'accélération et de décélération, une solution serait de d'accélérer d'une manière constante pendant la moitié du trajet et de décélérer avec une décélération constante sur l'autre moitié du trajet. Le profil de vitesse est ainsi triangulaire et le profil de position est parabolique.



Hypothèses de calcul du profil:

- Accélération Acc_max
- Décélération identique à l'accélération
- La longueur du trajet Δx

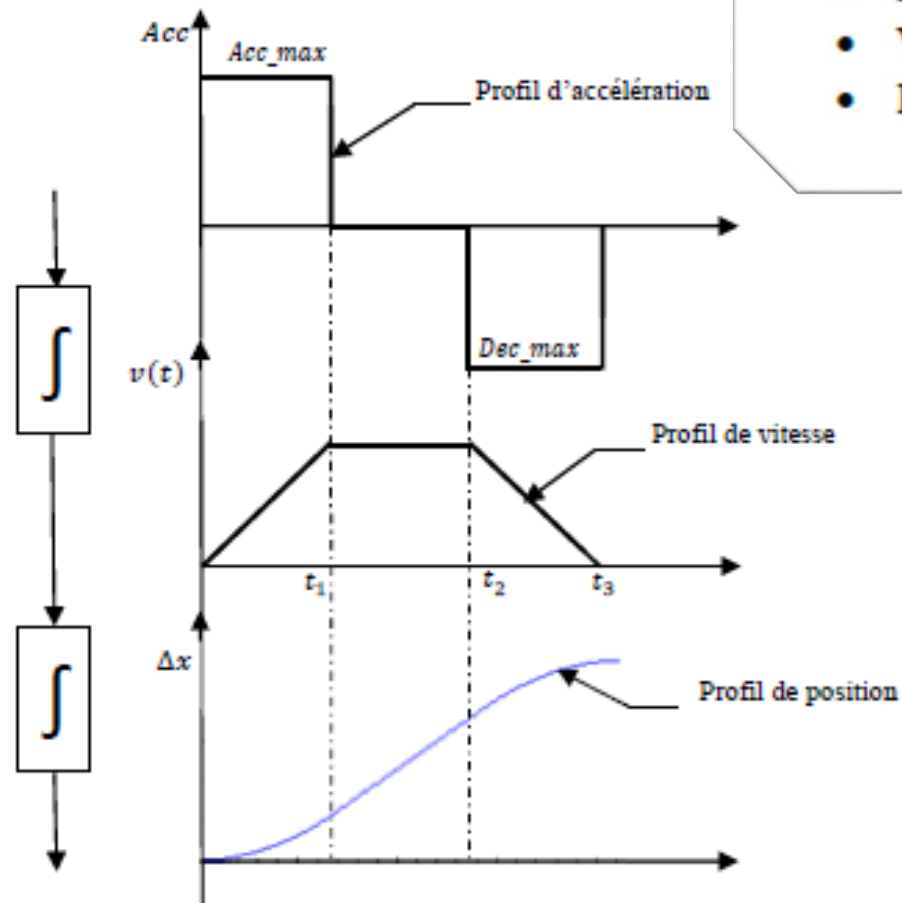
Inconnu: t_f

Génération de la consigne :

$$x(t) = \begin{cases} x_0 & t \leq 0 \\ x_0 + Acc_max \frac{t^2}{2} & 0 < t < \frac{t_f}{2} \\ \frac{x_1 - x_0}{2} + V_{max} \left(t - \frac{t_f}{2} \right) - \frac{Dec_max}{2} \left(t - \frac{t_f}{2} \right)^2 & \frac{t_f}{2} \leq t < t_f \\ x_1 & t \geq t_f \end{cases}$$

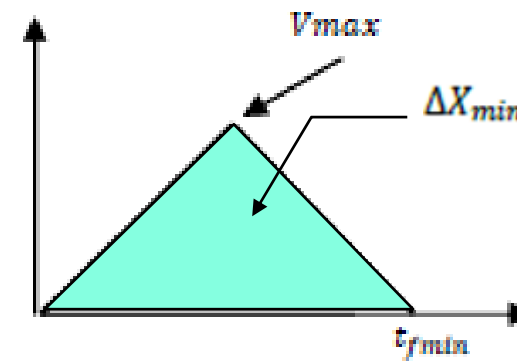
Ce profil permet de réaliser le trajet en un minimum de temps tout en ayant une accélération et décélération constantes. Néanmoins, l'inconvénient est que la vitesse de pointe peut être très élevée au cas où le parcours est long. Le profil trapézoïdal en vitesse est alors préconisé car il permet de limiter la vitesse à une vitesse maximale définie dans les hypothèses.

Le profil de vitesse trapézoïdale :

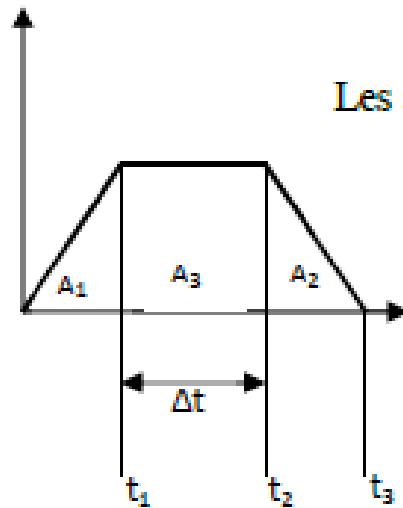


Hypothèses de calcul du profil :

- Accélération Acc_max .
- Décélération identique à l'accélération.
- Vitesse max V_{max} .
- La longueur du trajet Δx



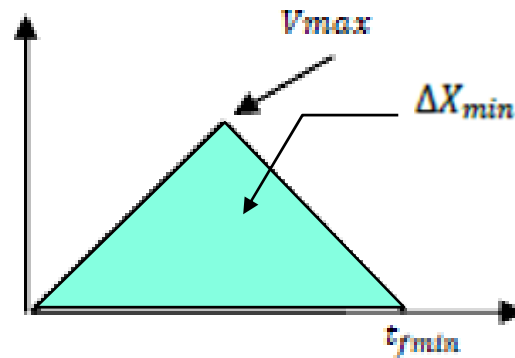
Si $\Delta X \geq \Delta X_{min}$, V_{max} est alors atteinte.



Les aires A_1 , A_2 et A_3 sont données comme suit:

$$\begin{cases} A_1 + A_2 = \Delta S_{min} \\ A_3 = \Delta S - \Delta S_{min} \end{cases}$$

$$\Rightarrow \begin{cases} t_1 = \frac{t_{fmin}}{2} = \sqrt{\frac{\Delta X_{min}}{Acc_{max}}} \\ t_2 = t_1 + \Delta t \\ t_3 = t_2 + \frac{t_{fmin}}{2} = 2 \sqrt{\frac{\Delta X_{min}}{Acc_{max}}} + \frac{\Delta x - \Delta X_{min}}{V_{max}} \end{cases}$$



2^{ème} cas :

Si $\Delta X < \Delta X_{min}$ alors V_{max} ne sera pas atteinte et le profil sera purement triangulaire. La vitesse maximale devra être adaptée pour réaliser le trajet à parcourir.

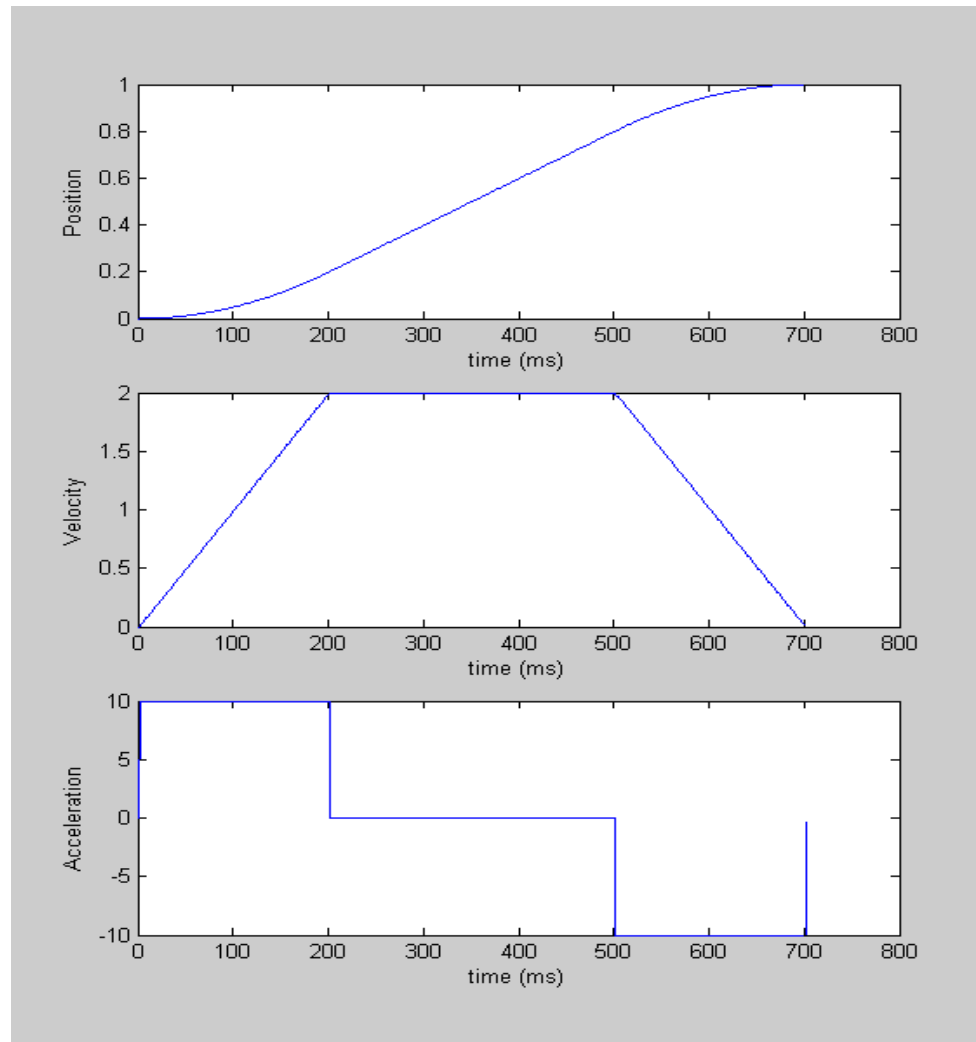
$$V_{max_new} = \sqrt{\Delta x * Acc_max}$$

$$t_1 = t_2 = t_3/2 = t_f/2 \quad \text{et} \quad t_f = 2 \sqrt{\frac{\Delta x}{Acc_max}}$$

Profil trapézoïdal (suite)?

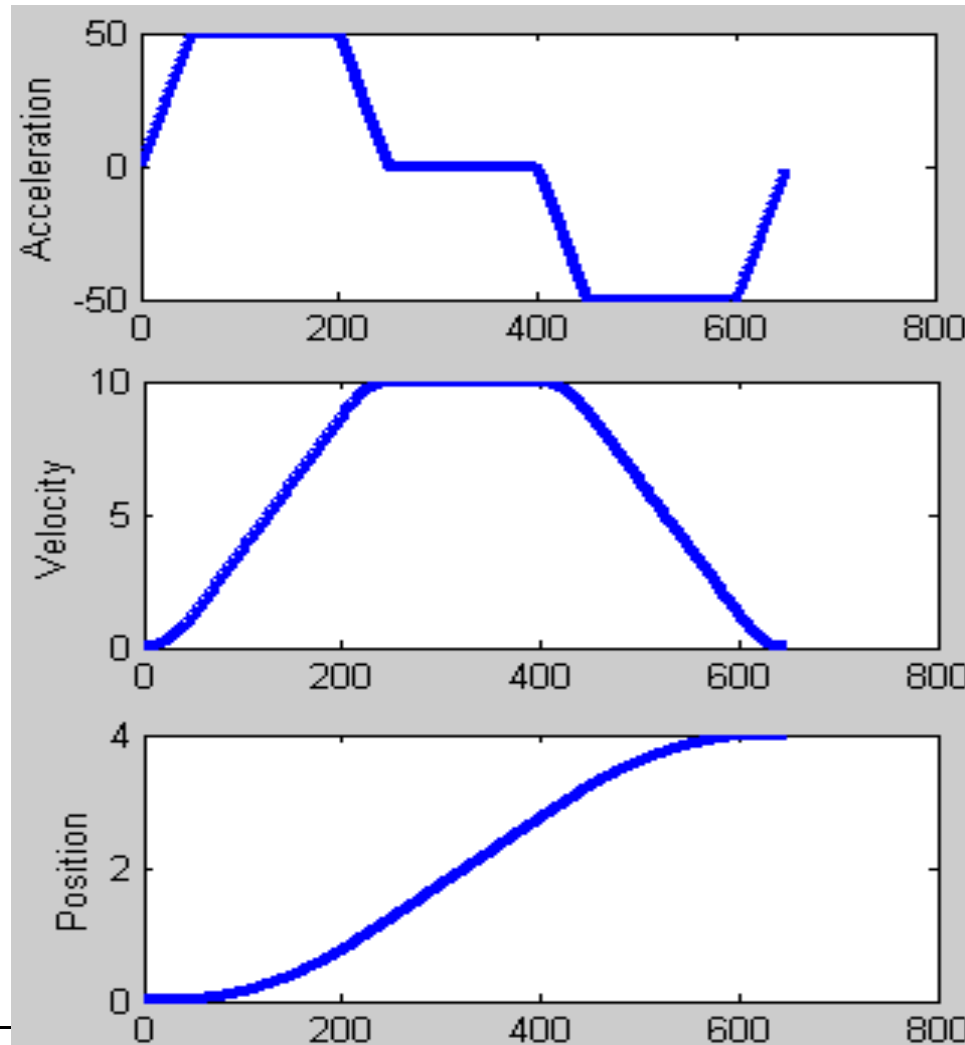


Le profil le plus connu est le *profil trapézoïdal en vitesse* (appelé aussi Bang-Bang en accélération et parabolique en position):

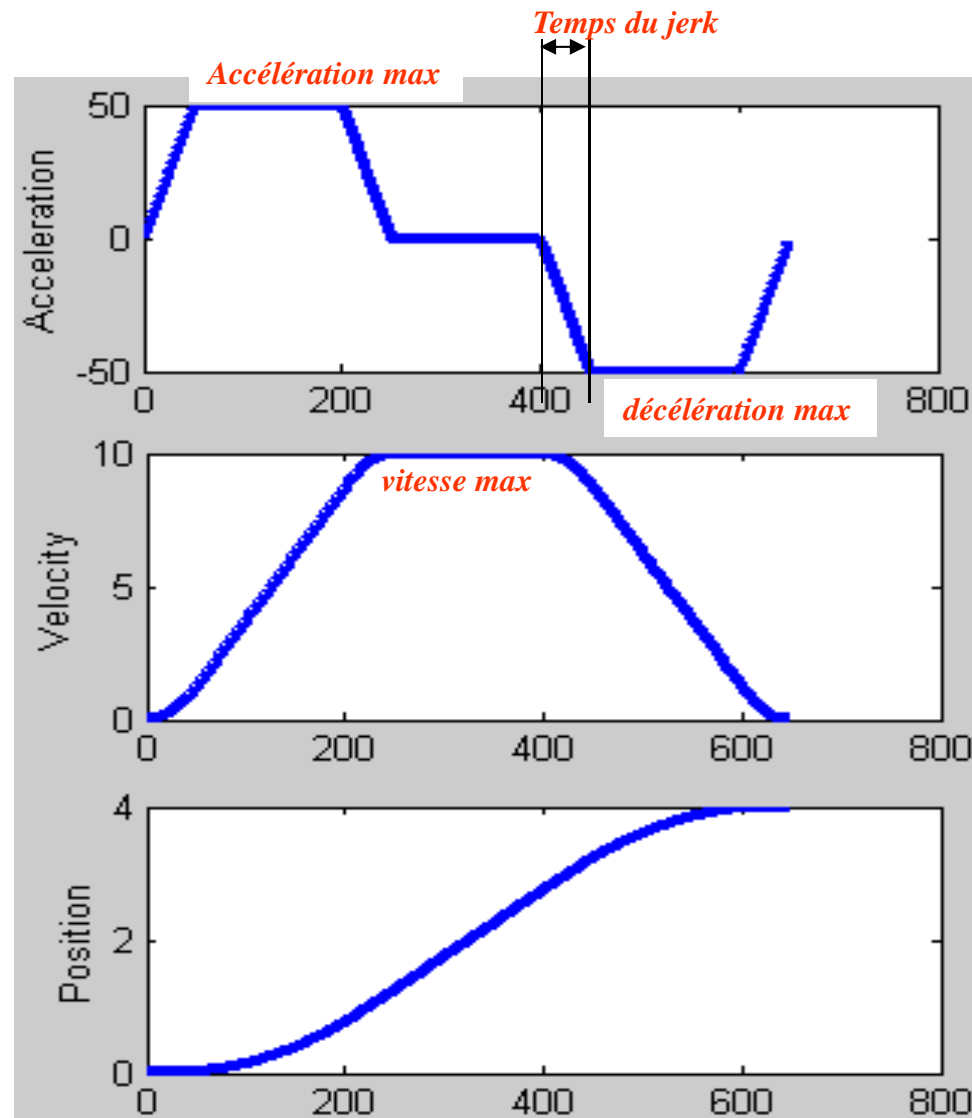


Autre profil : Trapézoïdal en accélération

↪ Profil plus lisse pour la mécanique est le profil trapézoïdal en accélération (appelé parabolique en vitesse):



Trapézoidal en accélération - paramètres



Génération multiaxes: Passage de la génération curviligne à la génération du trajet.



L'interpolation

- 1- Génération d'un segment de droite – interpolation linéaire
- 2- Génération d'un cercle et d'un arc de cercle

La synchronisation des axes



I - L'interpolation

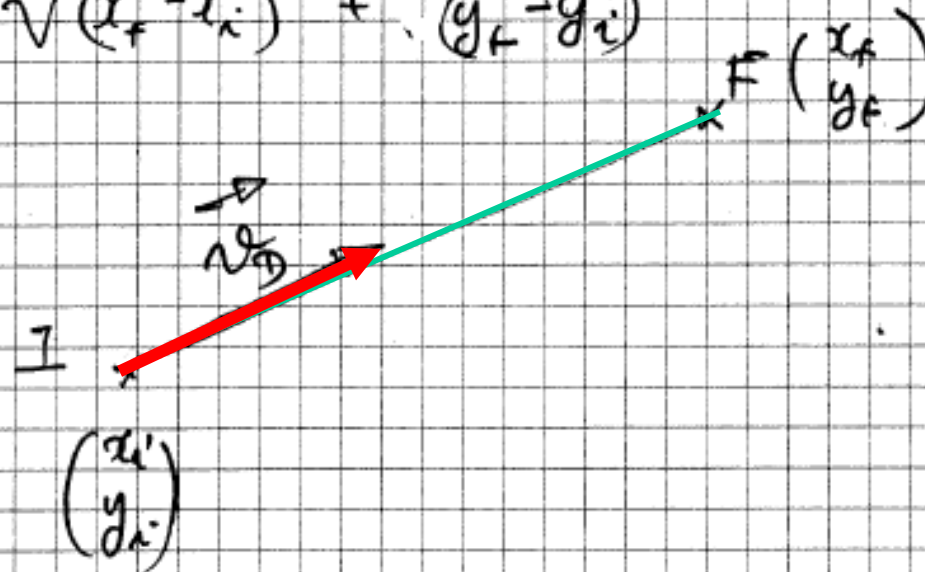
- 1- Génération d'un segment de droite – interpolation linéaire
- 2- Génération d'un cercle et d'un arc de cercle

→ 1^{er} cas : ligne droite depuis point I (x_i, y_i)
à point F (x_f, y_f)

le trajet à parcourir est

$$\Delta S = \sqrt{\Delta x^2 + \Delta y^2}$$

$$= \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$$



soit \vec{v}_s le vecteur directeur le long du parcours.

$$\vec{v}_s = \begin{pmatrix} \frac{x_f - x_i}{\Delta s} \\ \frac{y_f - y_i}{\Delta s} \end{pmatrix} = \begin{cases} v_{sx} \\ v_{sy} \end{cases}$$



La première opération à faire est de générer le profil de position $s(t)$ de longueur Δs .



La deuxième opération consiste à projeter le profil sur les coordonnées $x(t)$ et $y(t)$ correspondantes.

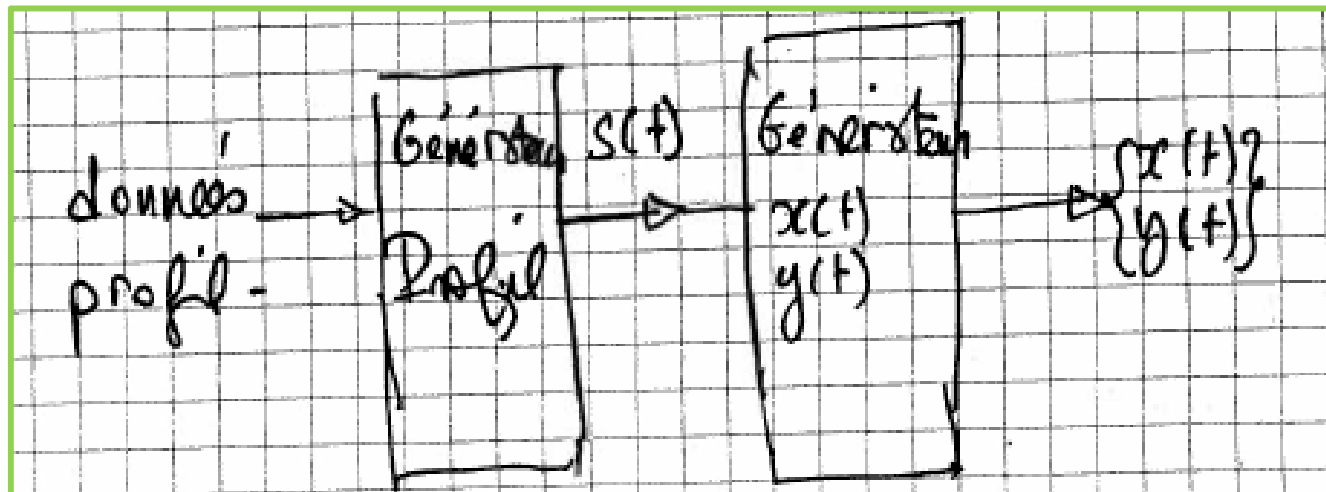
$$x(t) = x_i + \frac{\partial x}{\partial s} \cdot s(t)$$

$$y(t) = y_i + \frac{\partial y}{\partial s} \cdot s(t)$$

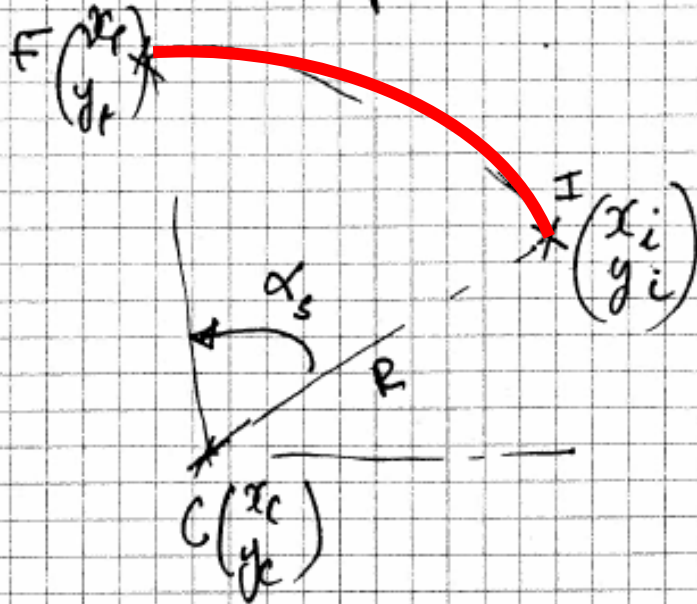
avec

$$x(t=0) = x_i \quad , \quad x(t=t_f) = x_f \quad .$$

$$y(t=0) = y_i \quad ; \quad y(t=t_f) = y_f \quad .$$



cas d'une interpolation circulaire



Dans ce cas il s'agit de réaliser une trajectoire circulaire (arc de cercle) depuis le point I au point "F". C'est le centre du cercle considéré. R est le rayon de ce cercle.

soit α l'angle à parcourir variant de 0 à α_s .



• La première étape consiste à générer le profil de l'angle $\psi(t)$ grâce à un choix approprié $\{ \text{trapézoïdal en vitesse ou trapézoïdal en accélération} \}$.

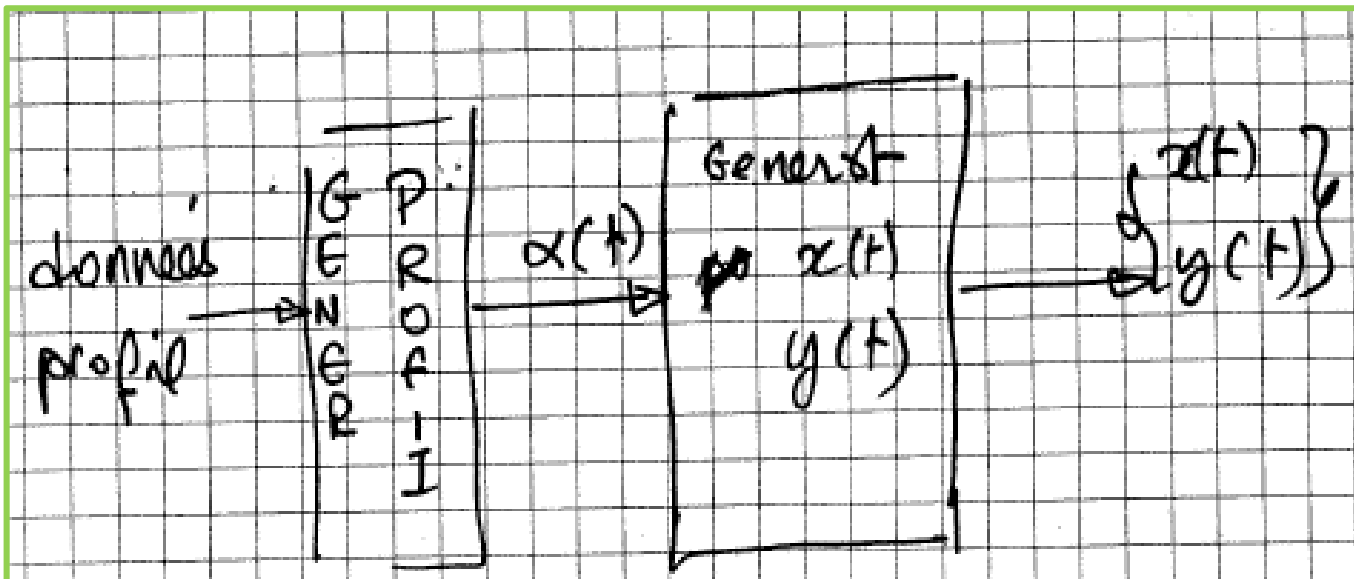


• la deuxième opération consiste à déduire les coordonnées $(x(t), y(t))$ le long du parcours circulaire.

$$x(t) = x_c + R \sin(\alpha(t))$$

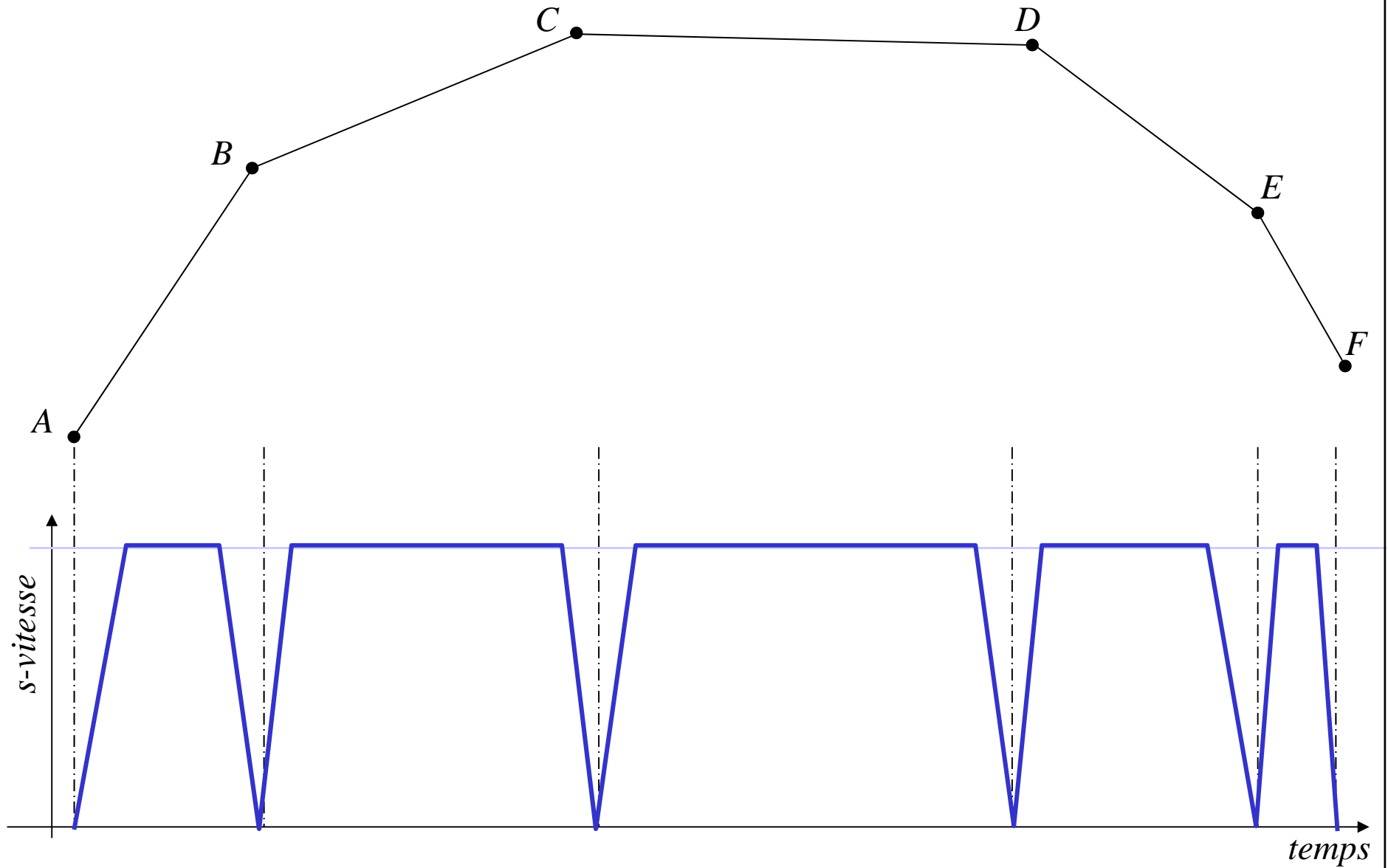
$$y(t) = y_c + R \cos(\alpha(t))$$

$$\begin{cases} x(t=0) = x_i ; & x(t=t_f) = x_f ; \\ y(t=0) = y_i ; & y(t=t_f) = y_f ; \end{cases}$$

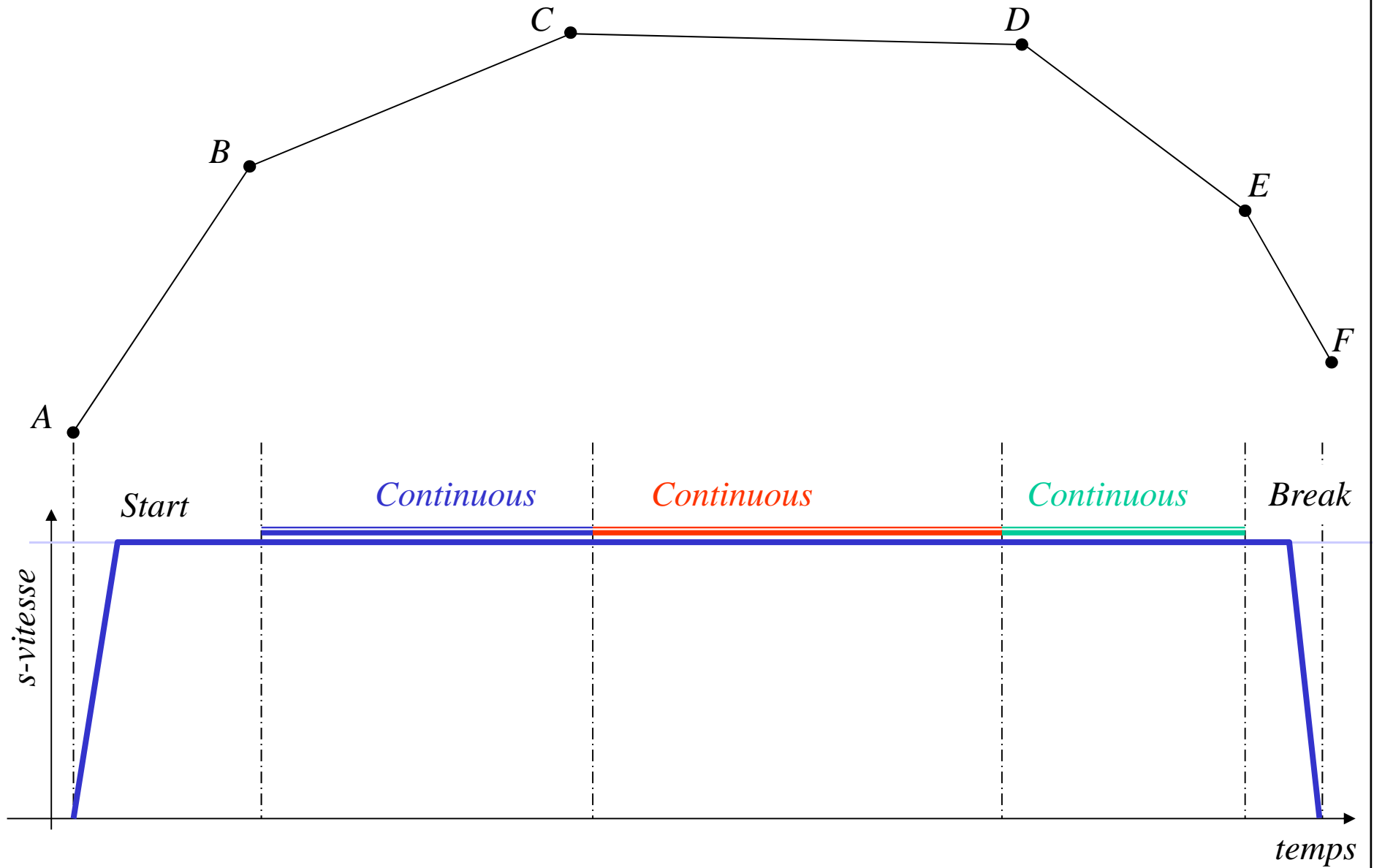


Raccord de trajectoires

Trajet en Point à Point

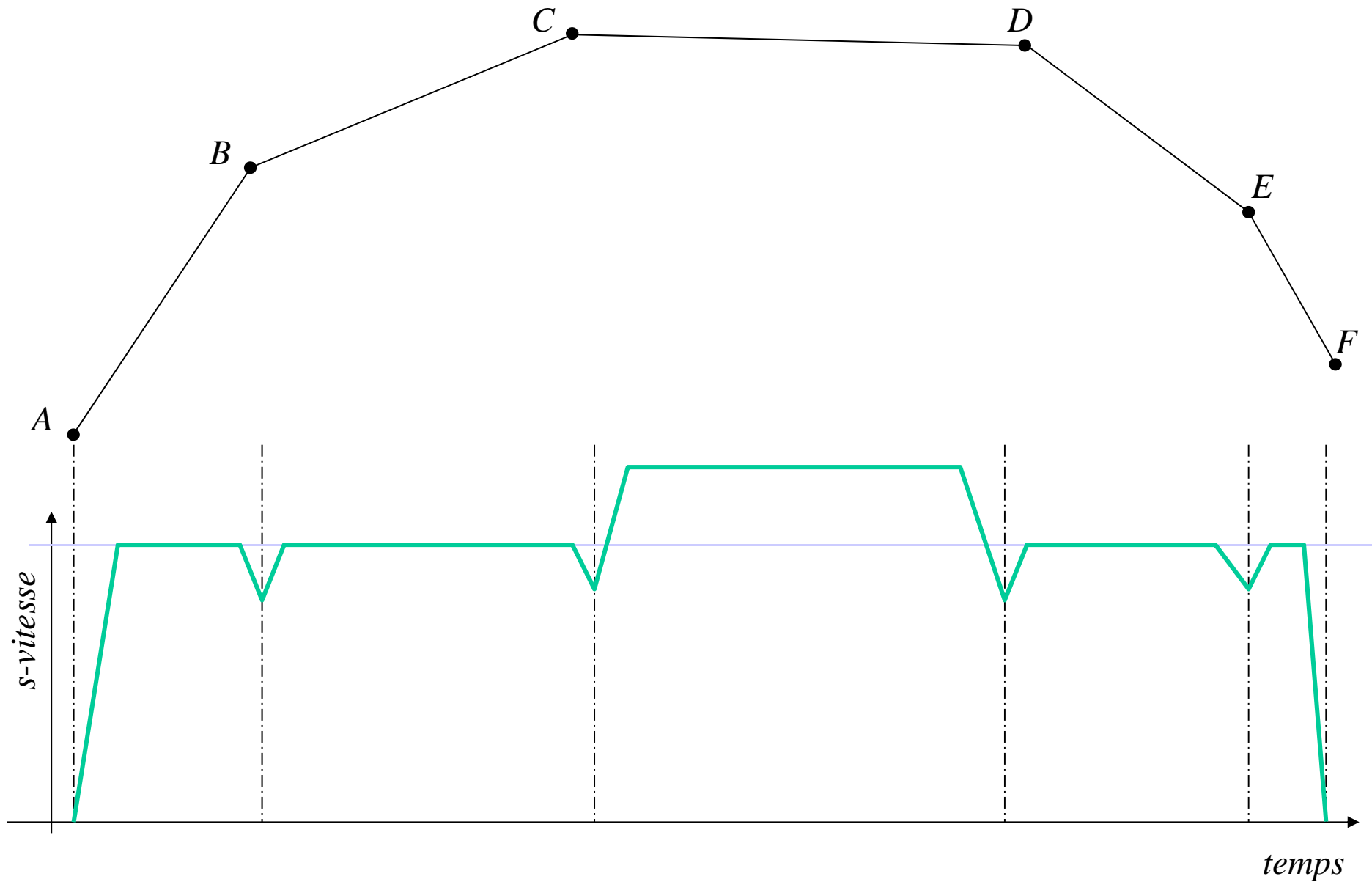


Trajet en Pont à Point (Continu)

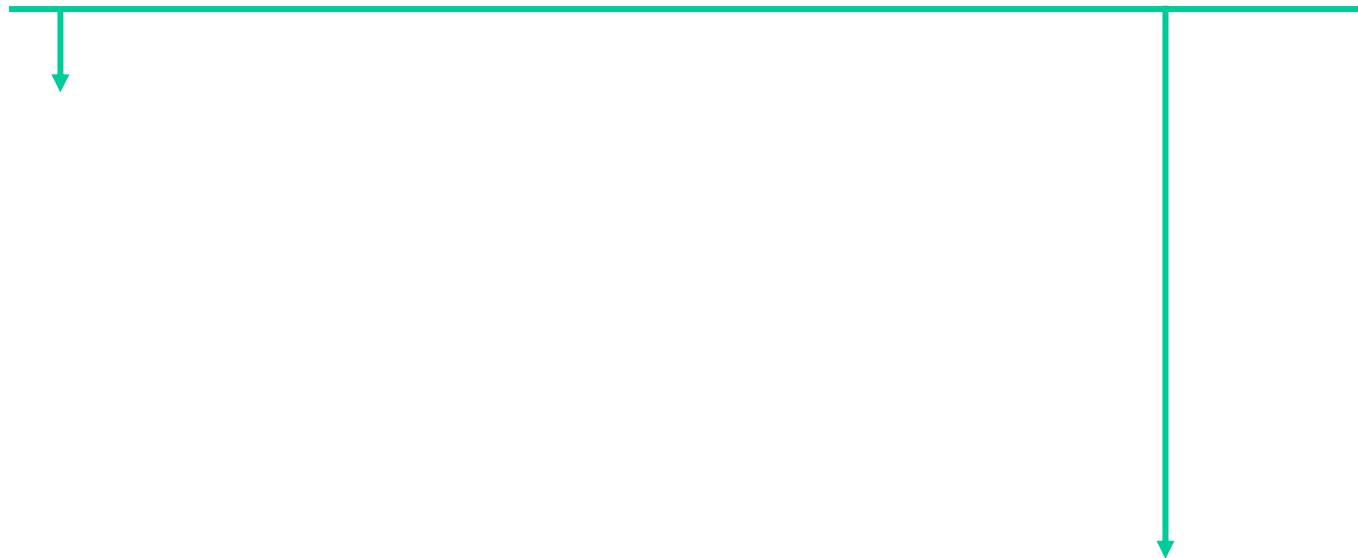


Quelles accélérations générées dans les raccords?

Profils continus (suite) : Gestion des raccords



Génération multiaxes:
Passage de la génération curviligne à la
génération du trajet.



II- La synchronisation des axes

Synchronisation des axes

Pour certains axes, la dépendance géométrique n'existe pas.

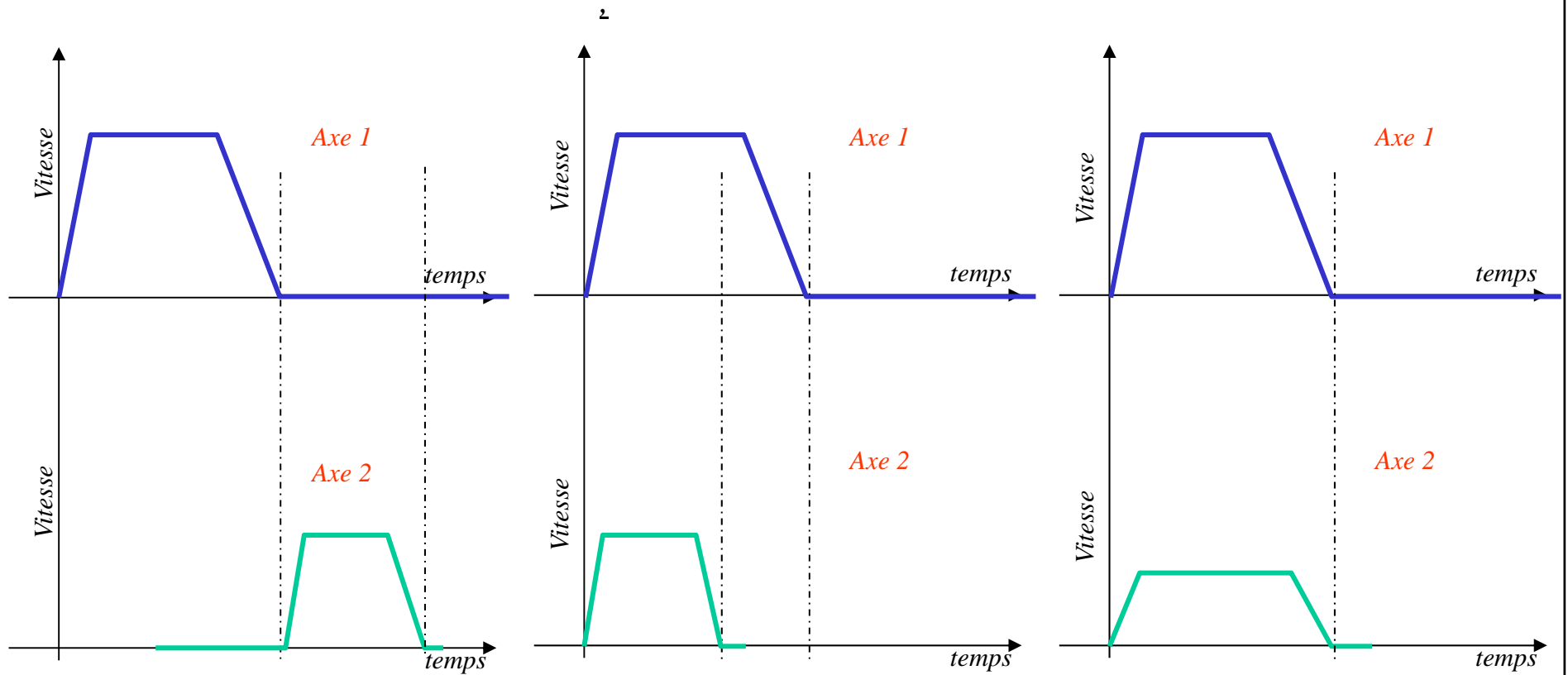
Exemples,

- deux axes rotatifs d'une machines,
 - deux axes linéaires sur une même direction,
- voir plusieurs axes d'une machine.

Dans ce cas, le mouvement d'une position 1 définis par $(p1_x1, p1_x2)$ vers une position 2 $(p2_x1, p2_x2)$, peut s'effectuer soit :

- en simultané axe par axe,
- en indépendant, ie chaque axe selon sa dynamique
- en synchronisé,

Synchronisation des axes (suite)



Différences avec l'interpolation ?

1- Mode x et y synchronisés –

Vitesse constante ---

Profil d'accélération ---

2- Mode articulaire -

Exemple 1:

G00 X12 Y3 ;	←	Déplacement rapide au point de coordonnées (X=12 et Y=3)
G00 X10 Y6 Z-5 ;	←	Déplacement rapide au point de coordonnées (X=12, Y=3, Z=-5)

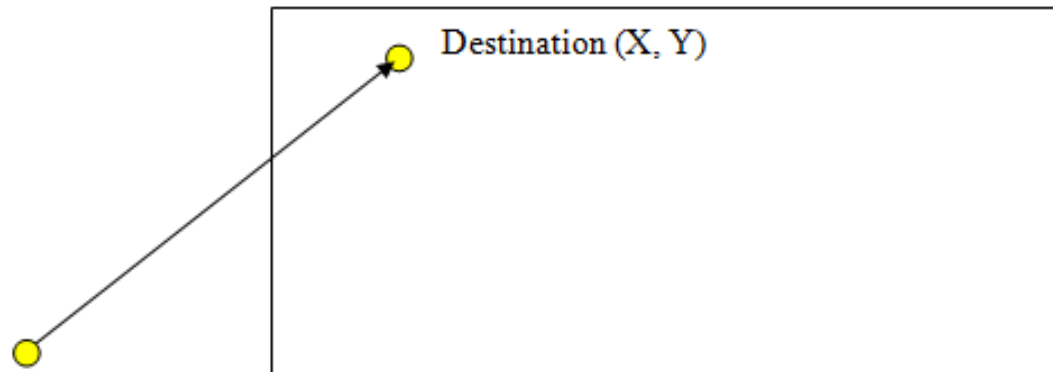


Figure 1 – Interpolation linéaire vers un point de coordonnées (X, Y)

Exemple 2:

G00 X12 Y3 ;	←	Déplacement rapide au point de coordonnées (X=12 et Y=3)
X10 Y6 Z-5 ;	←	Poursuite du déplacement rapide au point de coordonnées (X=12, Y=3, Z=-5)
...		
X15 Y7 ;	←	Poursuite du déplacement rapide au point de coordonnées (X=12, Y=3, Z=-5)

Remarque : Le code G est un code *modal*. Ceci est aisément montré par l'exemple précédent. La deuxième ligne (X10 Y6 Z-5 ;) ne contient pas de commande G. Elle est associée à la dernière commande se trouvant dans les lignes précédentes du programme. Dans cet exemple cette commande est la commande G00. Cette commande sera mémorisée jusqu'à ce qu'elle soit modifiée.

G01 Mouvement à vitesse contrôlée

La commande G01 permet un déplacement linéaire (interpolation linéaire) aux coordonnées indiquées avec une vitesse de déplacement contrôlée. Par vitesse contrôlée nous voulons dire une vitesse programmable par le code G via le paramètre de vitesses **F** (**F** comme Feedrate).

Exemple 1:

```
G01 X1.5 Y3 F300;
```

← Déplacement au point de coordonnées (X=1.5, Y=3) à 300 mm/s

Exemple 2:

```
G01 X1.5 Y3 F300;
```

```
X10 Y6 Z-5 ;
```

```
...
```

```
X15 Y7 ;
```

← Déplacement rapide au point de coordonnées (X=12 et Y=3)

← Poursuite du déplacement rapide au point de coordonnées (X=12, Y=3, Z=-5)

← Poursuite du déplacement rapide au point de coordonnées (X=12, Y=3, Z=-5)

Le paramètre F peut être modifié soit

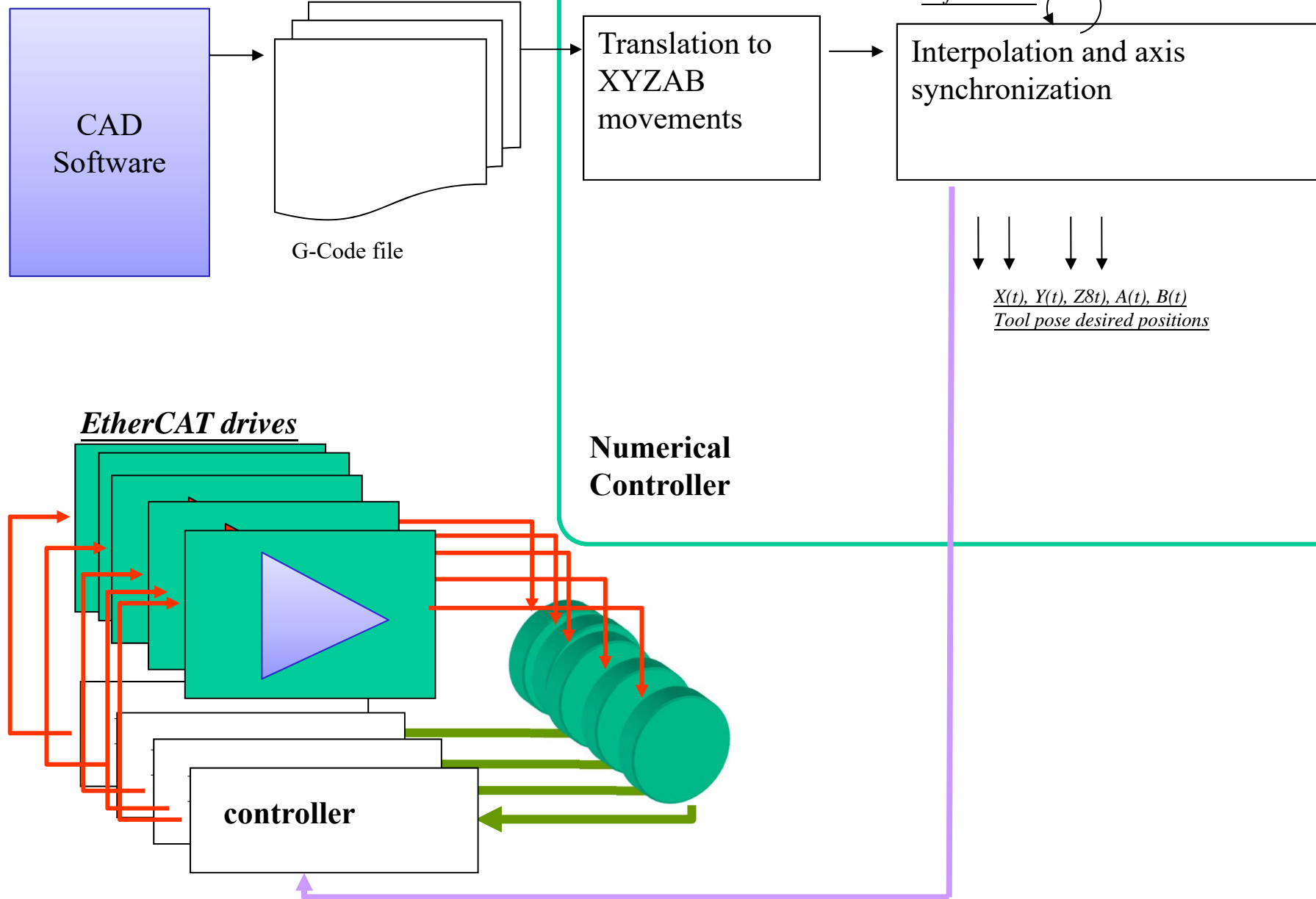
- o sur la même ligne de commande de mouvement.

```
G01 X1.5 Y3 F300 ;  
X10 Y6 Z-5 ;  
...  
X15 Y7 ;
```

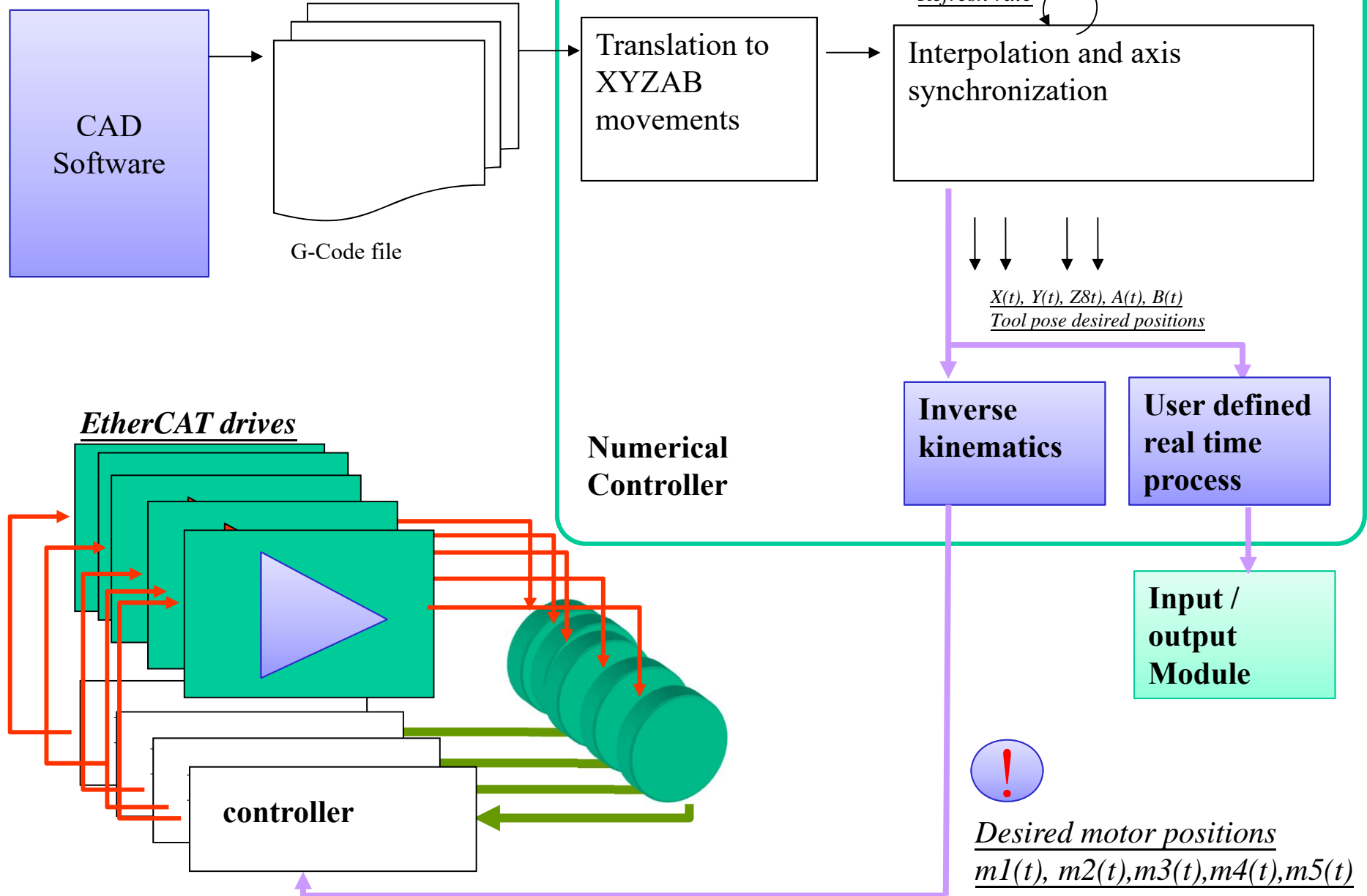
- o sur une ligne de commande à part en utilisant le paramètre de vitesse d'avance F en écrivant.

```
F300 ;  
G01 X1.5 Y3;  
X10 Y6 Z-5 ;  
...  
X15 Y7 ;
```

Exemple d'implémentation 1



Exemple d'implémentation 2




Nous allons voir:

 le matériel,

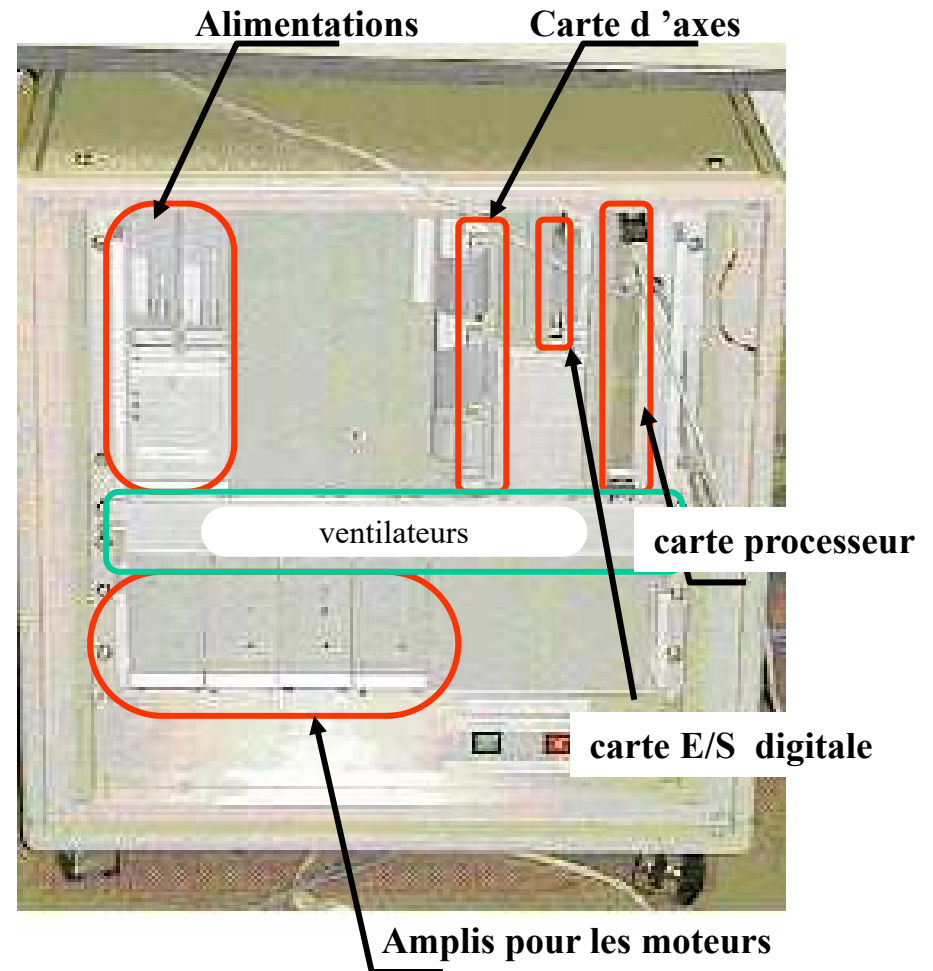
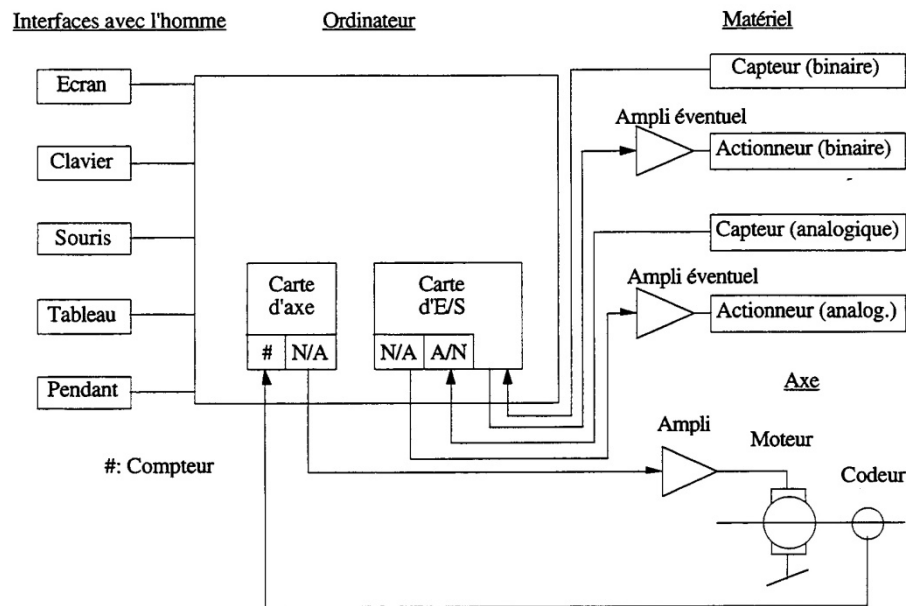
 les OS's,

 l'application de contrôle: architecture,

 quelques comparaisons : bus, OS, cartes, systèmes,...

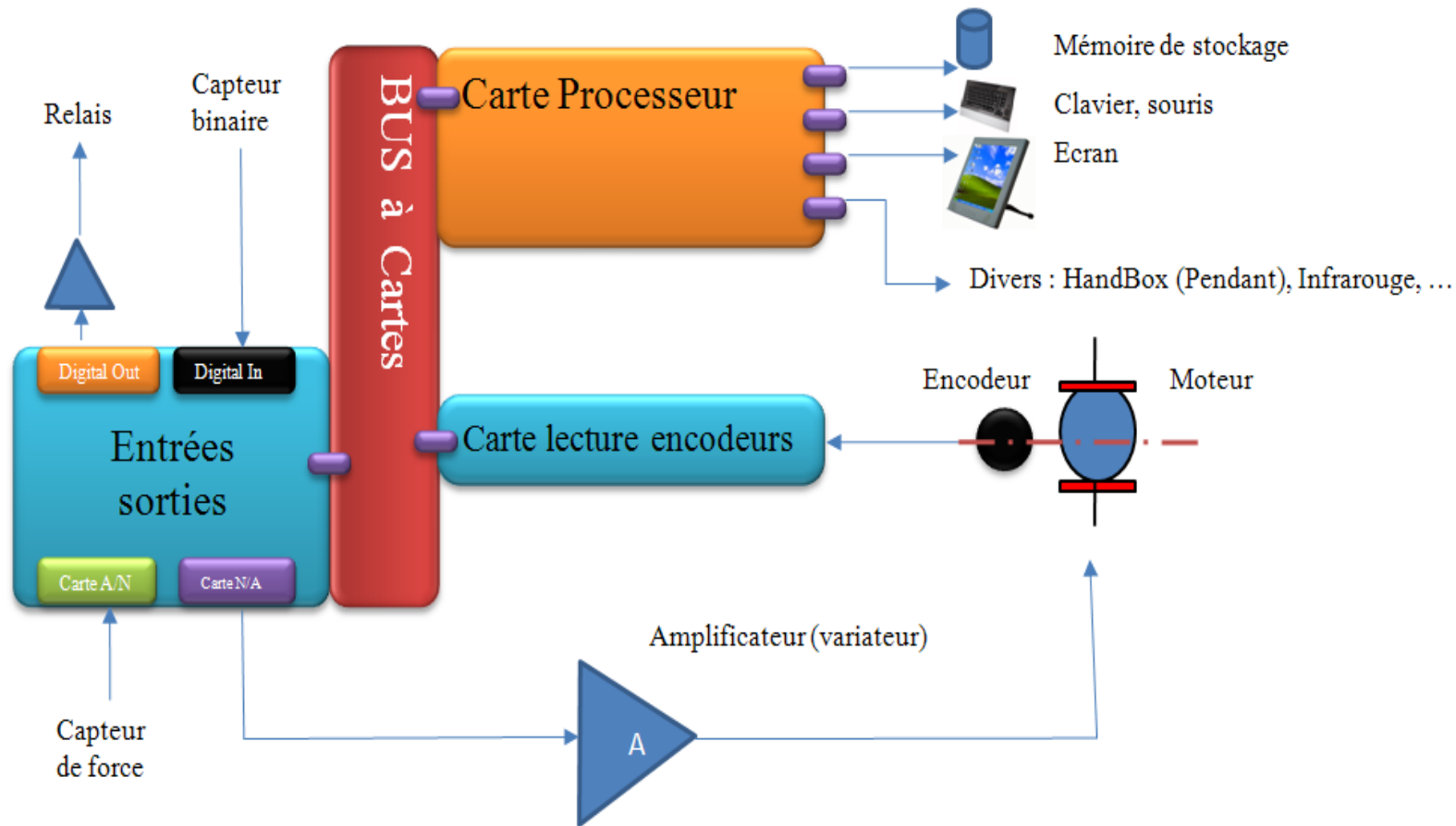
Matériel:

Eléments matériels d'une armoire de commande



Matériel:

Eléments matériels d'une armoire de commande



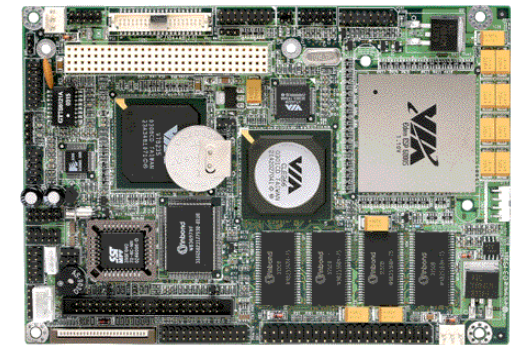
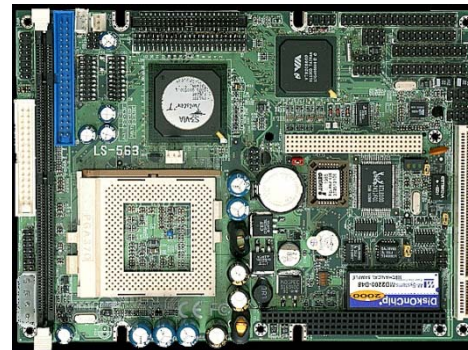
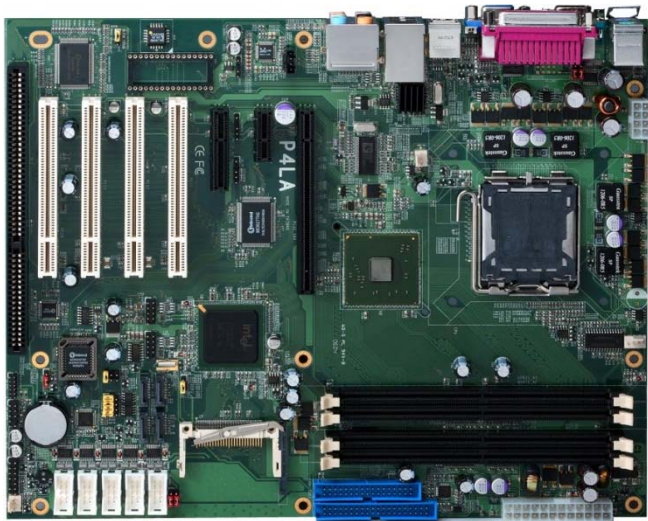
Intelligence: Follow up!

PC technology

✓ Motherboards

✓ Micro Motherboards

✓ Only Cpu boards



Drives de moteurs et Bus de terrain

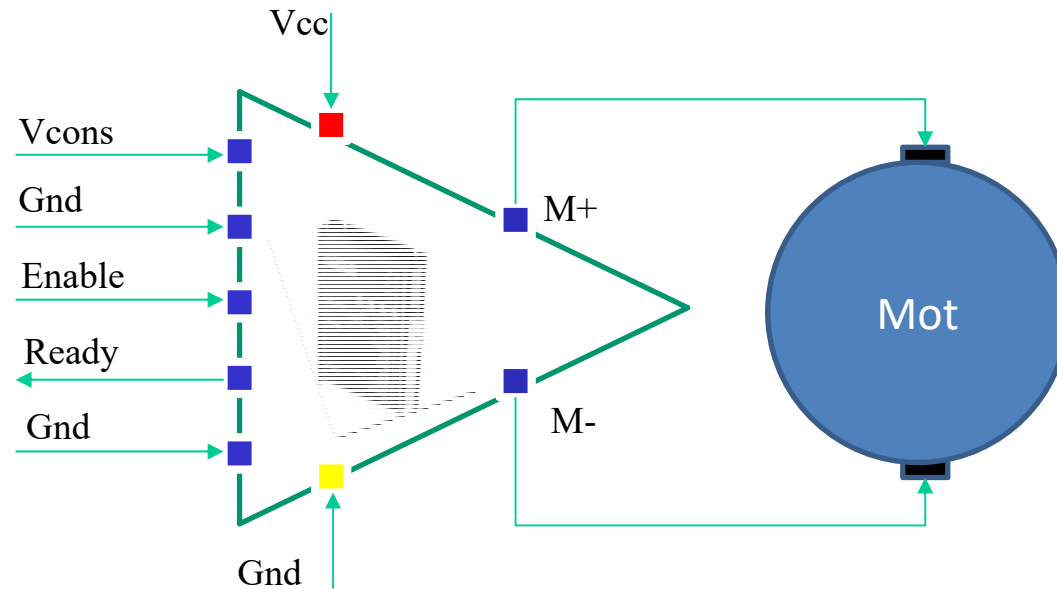
– Par analogie avec quoi ?

- Technologie conventionnelle : *consignes analogiques*
 - *De quoi ?*

- Bus de terrain – *Communication numérique*

Technologie analogique, exemple:

102

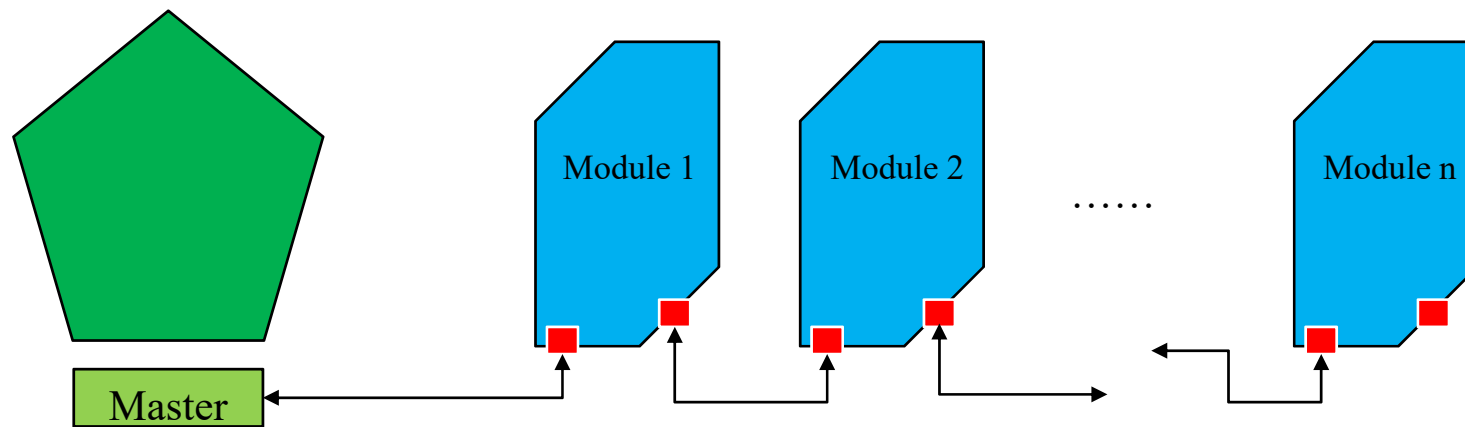


Exemple de câblage d'un servo-amplificateur Maxon (description dans tableau ci-dessous)

Signal	Type	Valeur	Description
Vcons	Analogique	-10V à 10V	Consigne de couple ou de vitesse selon configuration du drive
Gnd	Analogique	0V	Masse
En	Tout ou Rien	5V	Activation de l'ampli (Enable)
Ready	Tout ou Rien	5V	Etat de l'ampli (Ok ou non)
M+, M-	Analogique	--	Bornes d'un moteur à balais
Vcc	Analogique	50V/5A	Alimentation à courant continu du drive

Exemple simple de communication digitale:

Daisy chain topology



Bus industriels * Bus de terrain * Field busses

1- CANOPEN (1Mbit/s)

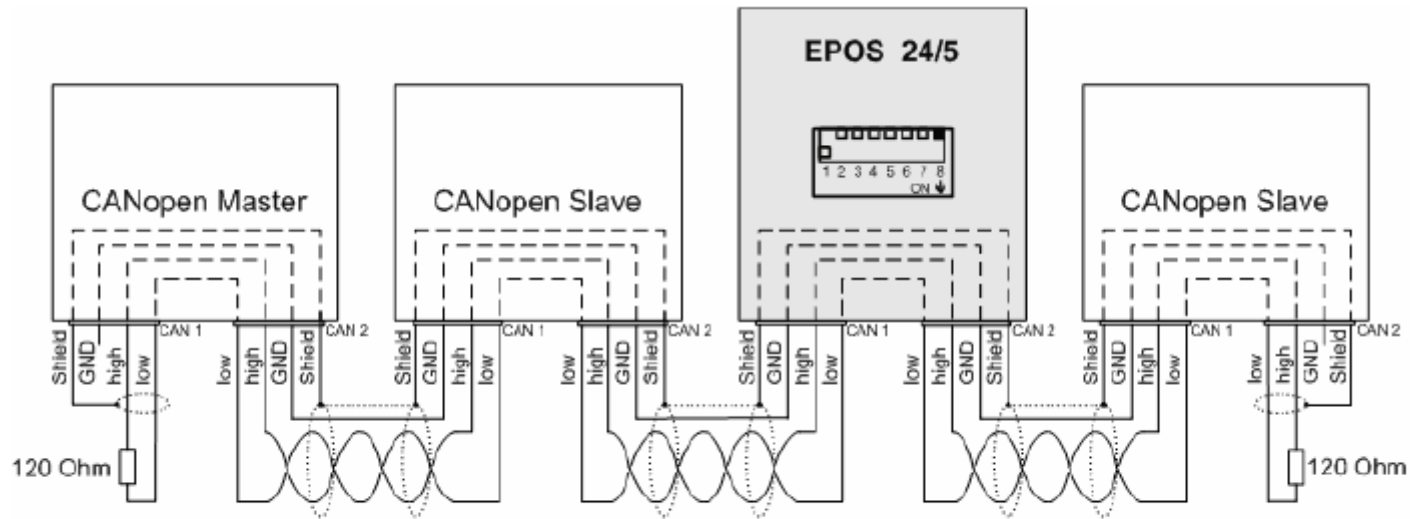


Figure 51: EPOS 24/5 without CAN-Bus termination

Bus industriels * Bus de terrain * Field busses

1- PROFIBUS (9.6 kbit/s to 12 Mbit/s)

This fieldbus is intended to interconnect plenty of devices of different types (e.g. sensors, drives, valves etc.). It permits the dialogue between equipment of **different manufacturers** without using highly specialized interfaces. Its universality and openness are assured by European and International standards. It is suitable for data transmission which require reflex actions and extremely short reaction times as well as transmission of high quantities of data.

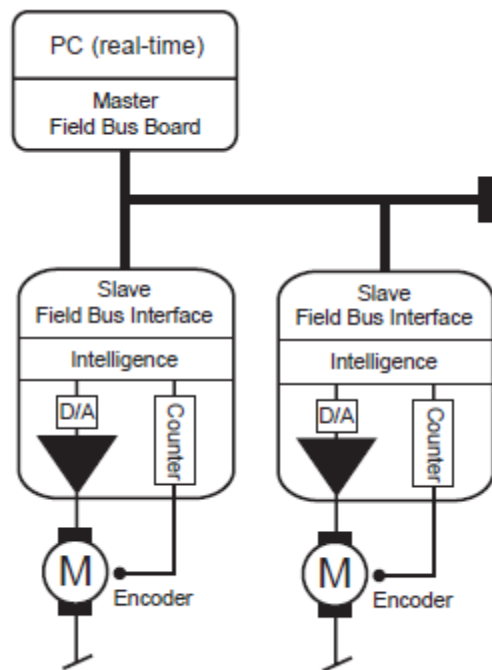


Figure 5: *The Control Hardware. The figure shows the 5 drives of our prototype. They contain the PROFIBUS-interface, the control-"intelligence" and the amplifiers.*

Bus industriels * Bus de terrain * Field busses

1- Ethercat

Transmission Rate: 2 x 100 Mbaud (Full-Duplex)

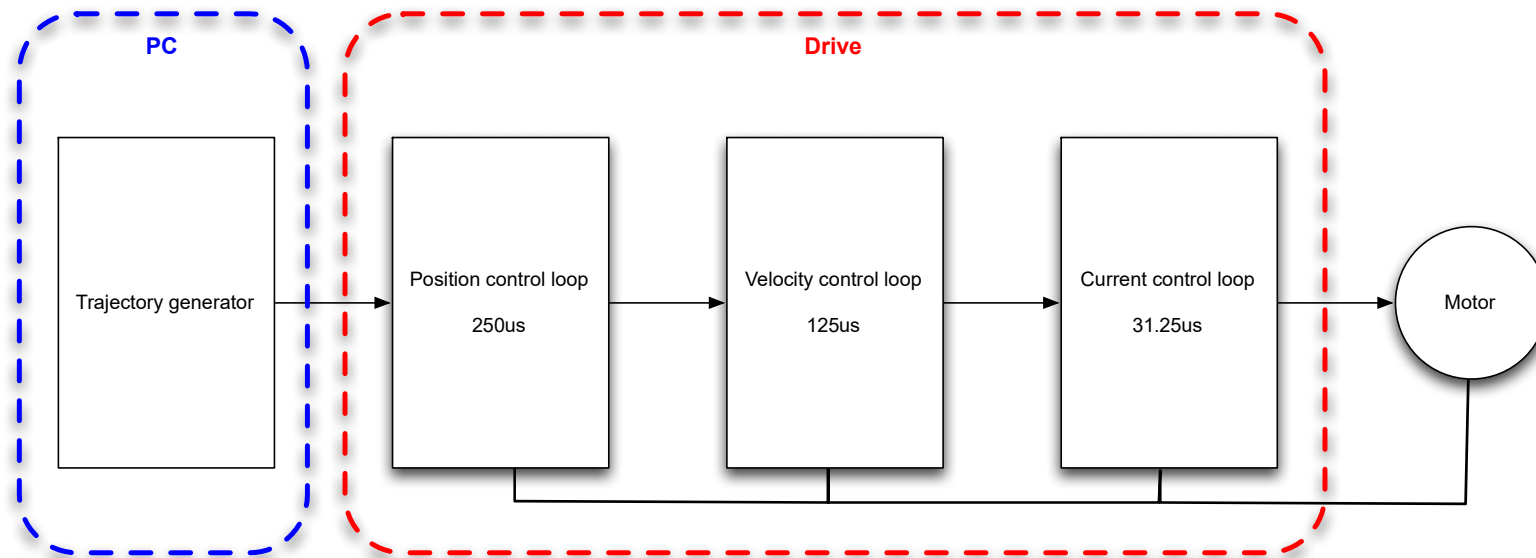
Update Times:

- 256 digital I/O in 11 μ s
- **1000 digital I/O** distributed to 100 nodes in **30 μ s = 0.03 ms**
- 200 analog I/O (16 bit) in 50 μ s, 20 kHz Sampling Rate
- **100 Servo-Axis** (each 8 Byte IN+OUT) in **100 μ s = 0.1 ms**
- 12000 digital I/O in 350 μ s



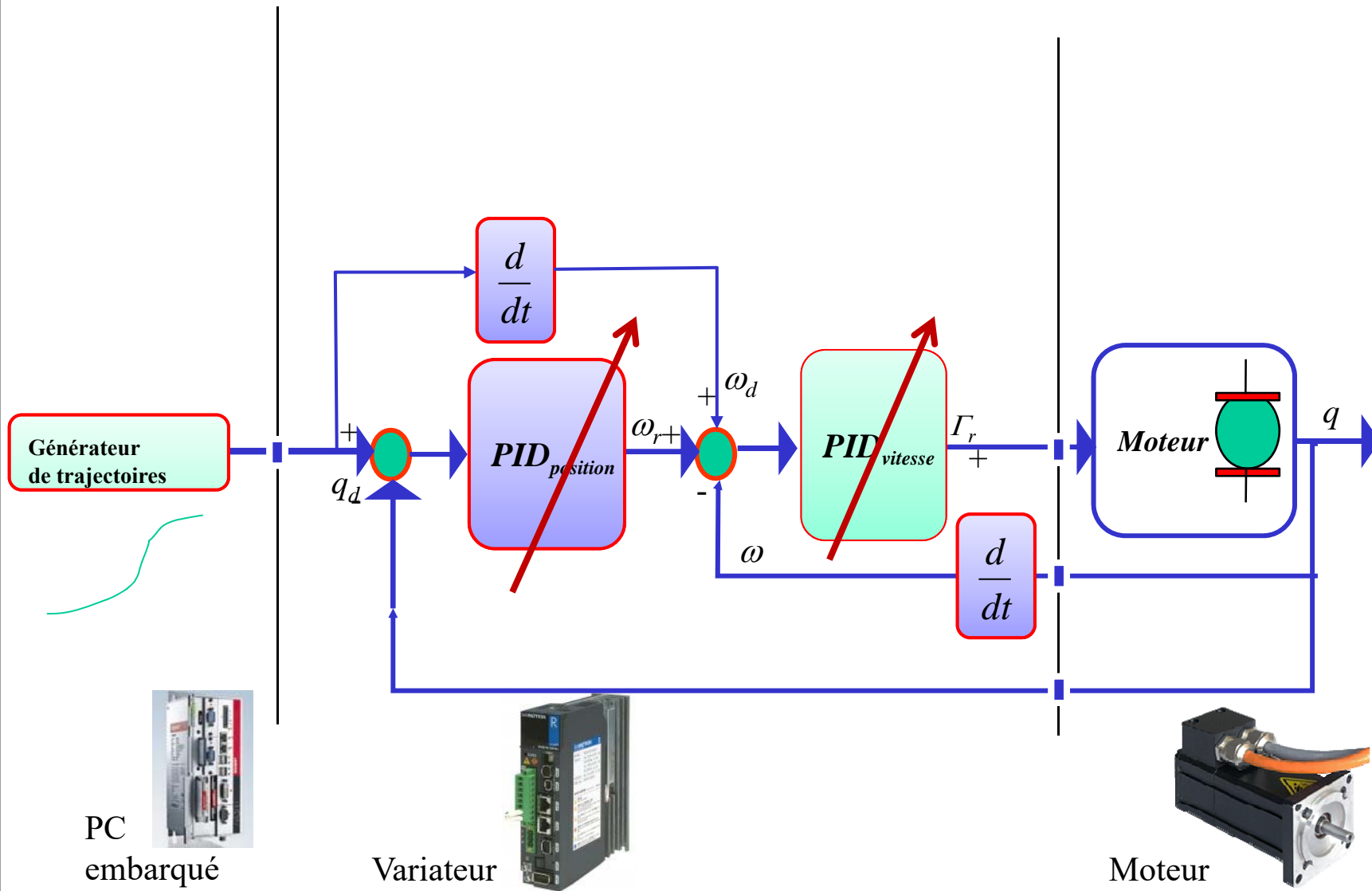
Commande cascadée par bus de terrain

The position and the velocity control loop can be placed either on the PC or on the drive. However, with the drive capacities, the most interesting configuration is as follows :

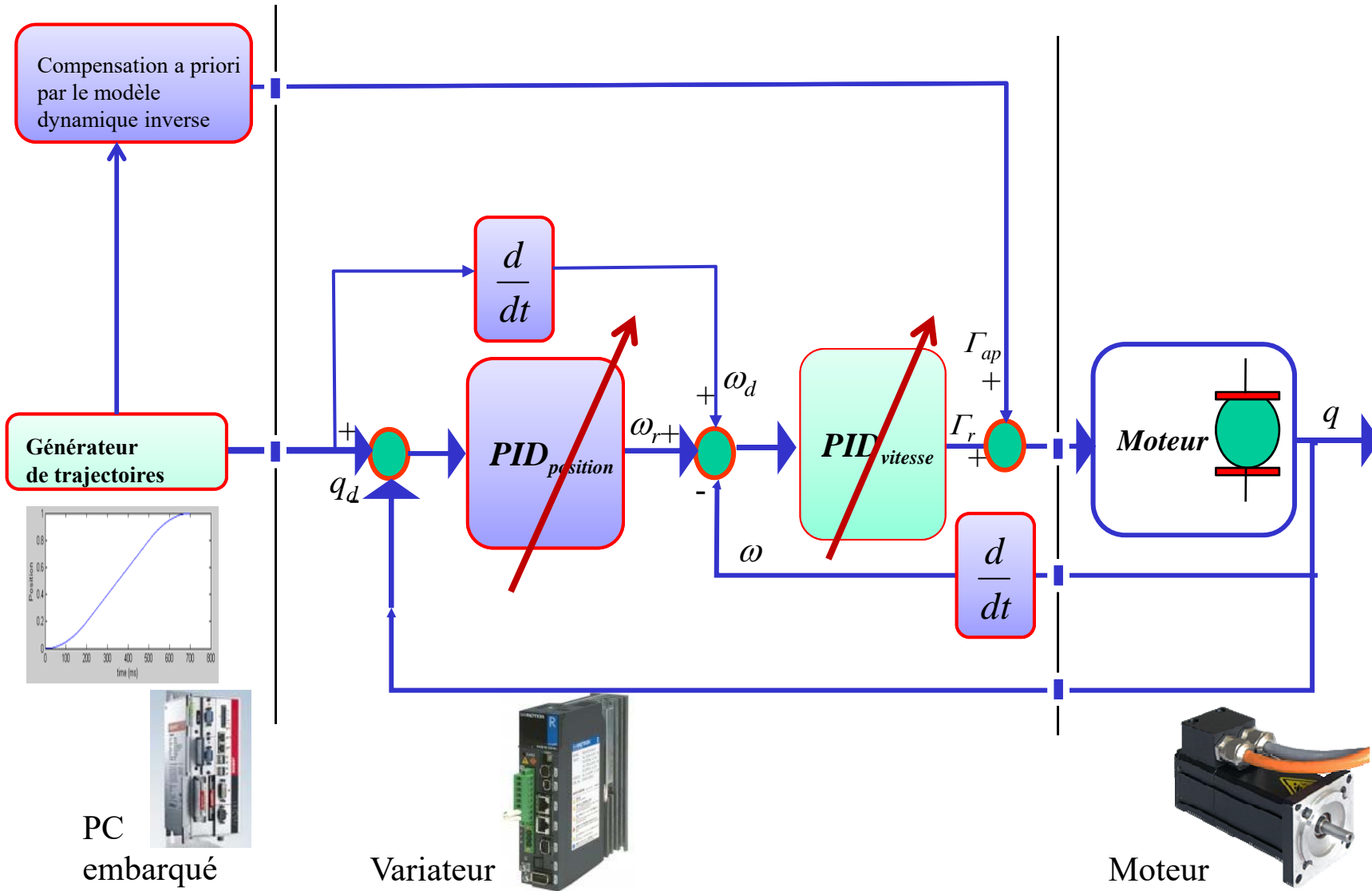


Moreover, the drive is capable to interpolate the command positions, in order to reduce the control discontinuities (steps).

Commande cascadée par bus de terrain



Commande par bus de terrain et a priori de couple



Question :

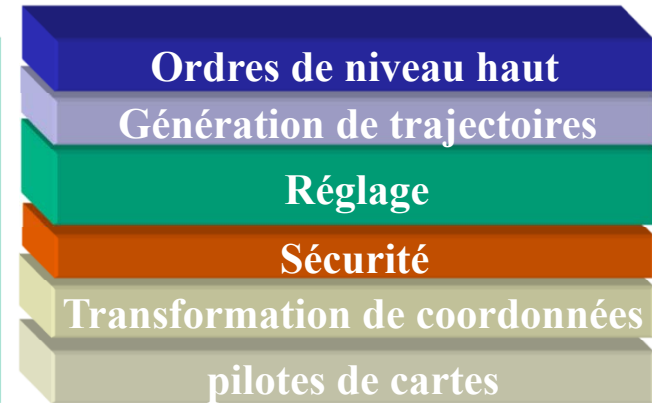
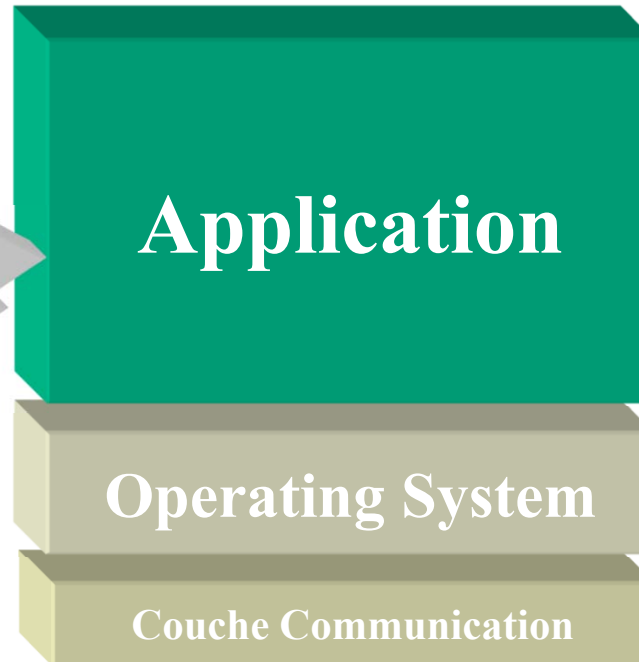
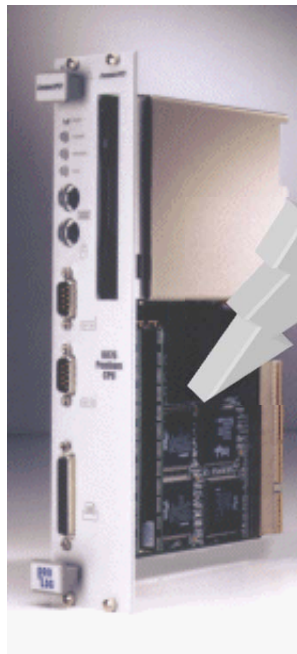
Quelles sont les avantages des bus de terrain?

1. Require le câblage.
2. Découpler la partie commande de la partie génération de consignes
3. Dissocier les métiers et les fournisseurs de solutions
4. Augmenter le nombre d'information à envoyer aux drives....

Aspect Logiciel



Application



Exigences temps réel d'un OS

Pour résumer

- ▣ multi-tâches supporté,
- ▣ évènements de synchronisation supportés,
- ▣ **déterminisme** temporel dans l'exécution des tâches,
- ▣ **déterminisme** temporel dans la prise en compte des interruptions,
- ▣ **déterminisme** temporel dans la manipulations des évènements.

Remarque: Windows n'est pas un système d'exploitation temps réel bien qu'il assure un déterminisme de l'ordre de la ms quand il n'est pas chargé. Certains l'utilise pour développer des applications **à temps réel non-critique** (en somme, qui n'obéissent pas aux exigences ci dessus!).

Mécanismes de communication temps réel

Un OS temps réel est un OS multi-tâches.

Une application temps réel doit être construite autour de plusieurs **tâches** ou **processus** communicants entre eux.

Les processus utilisent ce qu'on appelle des mécanismes IPC (Inter Process Communication mechanisms) pour se synchroniser et s'échanger des données.

Ces mécanismes sont:

- ***Les sémaphores binaires***: des objets qui peuvent être échangés et sont consommées ou fournies.
- ***Les sémaphores à compteur***: sont initialisées à un nombre donné et sont consommées ou fournies jusqu'à épuisement du stock.
- ***Les mutexes*** sont des sémaphores qui sont soit disponibles soit en utilisation par une ressource.
- ***Les mémoires partagées*** sont des zones mémoires.
- ***Les messageries*** pour échanger des chaînes de caractères.

Operating Systems!

RTOS

- ↳ Vx Works de Wind River Systems,
- ↳ pSos de Integrated Systems Inc.,
- ↳ Lynx Os de LynxOs Inc. pour systèmes embarqués et temps réel, devenu LynuxWorks
- ↳ Onx OS de QNX Software,
- ↳ OS9 OS de MicroWare,
- ↳ MS Windows CE de Microsoft,
- ↳ Autres développées par des universités et des laboratoires (Oberon(ETHZ),...)

Only embedded

- ↳ MS Windows XPe XPembedded de Microsoft pour systèmes embarqués,
- ↳ Linux
 - ↳ Linux tout court (from ...)
 - ↳ Linux Embedded(par ex. BlueCat de LynuxWorks(ex Lynx))

Extensions

- ↳ RTX de IntervalZero est une extension temps réel pour systèmes tournant sous OS microsoft,
- ↳ RTLlinux et RTAILib extensions temps réel pour Linux,
- ↳ Autres pour Linux,
- ↳ Autres pour MS Windows t.q. Hyperkernel, InTime,....

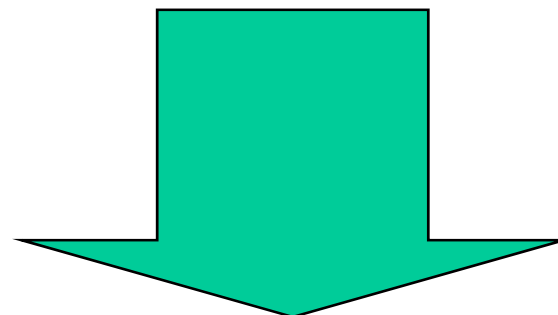
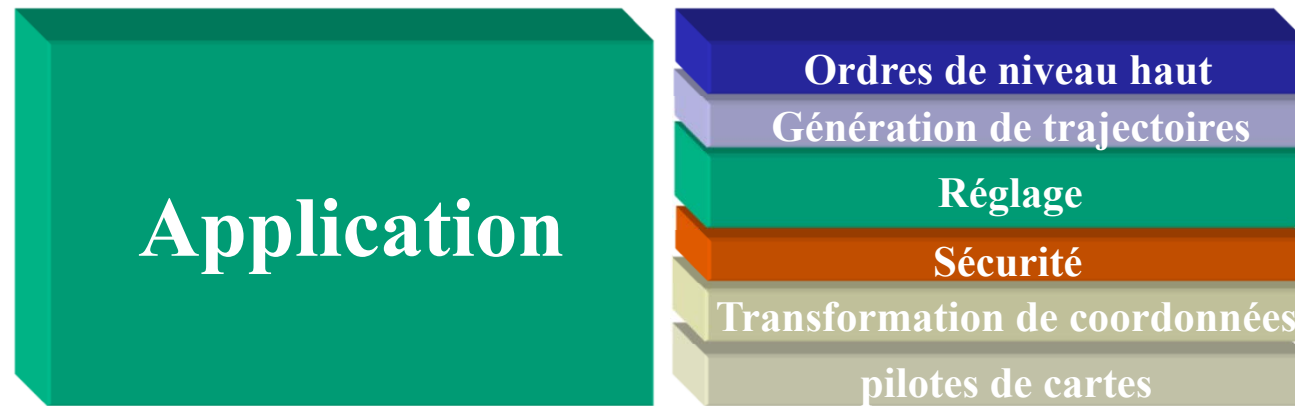


Extensions temps réel

Une extension temps réel **se greffe** sur un OS initial (XP , XPe, LINUX xx, ...).

- On utilise donc directement les outils préalablement existant sous ces OS's.
- On utilise les outils de debugging existants sous ces OS's.
- Des mécanismes de communication entre les tâches RT et non RT sont disponibles.
- Son prix est très accessible par rapport à un OS complet (par exemple 7000USD pour RTX contre 15,000 USD pour VxWorks)

Modules du contrôleur



Communication entre les modules

Quelle est la structure minimale d'un contrôleur de robot ?

- La bibliothèque d'instructions de **supervision** (`allezAupoint()`, `commence()`, `arrete()`, ...)
- La **boucle de réglage**, cette boucle est la plus importante elle assure une lecture capteurs, calcule couples moteurs et envois des commandes aux amplis; ...
- la **génération de la trajectoire**,

