

## ***Axe Instrumentation 2002-2003***

### ***Travaux pratiques d'Instrumentation Numérique***

<b>I</b>	<b>GENERALITES .....</b>	<b>3</b>
1.	Déroulement des TPs .....	3
2.	Contenu de ce document : .....	4
<b>II</b>	<b>INTRODUCTION A LABVIEW .....</b>	<b>5</b>
1.	Introduction.....	5
2.	Aide .....	5
3.	Exemple :.....	6
4.	Construction d'un premier instrument virtuel :.....	7
5.	Principes de Labview .....	8
6.	Techniques de débogage :.....	9
7.	Les structures (Palette Structures):.....	9
8.	Sous VIs .....	15
9.	Conclusion :.....	16
<b>III</b>	<b>L'ECHANTILLONAGE.....</b>	<b>17</b>
1.	La discrétisation dans le temps : .....	18
2.	La discrétisation de la grandeur mesurée : .....	18
3.	Le théorème de la moyenne :.....	19
<b>IV</b>	<b>TP N°1 REGULATION NUMERIQUE DE TEMPERATURE.....</b>	<b>22</b>
1.	Principe général:.....	22
2.	Dispositif, action et mesure :.....	23
3.	Identification du processus : .....	25
4.	La régulation PID :.....	27
<b>V</b>	<b>TP N°2: DETECTION SYNCHRONE.....</b>	<b>32</b>
1.	But:.....	32
2.	Préambule : Analyse fréquentielle & signaux périodiques.....	32
3.	Quelques outils disponibles : .....	34
4.	VIs à réaliser :.....	36
<b>VI</b>	<b>TP N°3 : LIAISONS NUMERIQUES / ETALONNAGE.....</b>	<b>40</b>
1.	Introduction :.....	40
2.	Préambule : Incertitude de mesure, régression linéaire .....	40
3.	Utilisation d'une liaison série RS232 : .....	45
4.	Première réalisation: multimètre .....	48
5.	Deuxième réalisation : balance numérique .....	48



# I. GENERALITES

## 1. *Déroulement des TPs*

Ces travaux pratiques ont une durée de 7 séances de 3 heures. **La première séance, d'initiation à l'environnement Labview, ne sera pas évaluée.**

**Les 6 autres** seront partagées en 3 TPs tournants, chaque TP de 2 séances faisant l'objet d'une évaluation séparée :

Séances	TP1 : Régul. température	TP2 : détection synchrone	TP3 : Liaison numérique, calibrage
2 et 3	A1, B1, A2, B2	A3,B3,A4,B4	A5,B5,A6,B6
4 et 5	A3,B3,A4,B4	A5,B5,A6,B6	A1, B1, A2, B2
6 et 7	A5,B5,A6,B6	A1, B1, A2, B2	A3,B3,A4,B4

Au cours de ces 3 TPs, les élèves construiront un ou des programmes (Ou V.I. = Virtual Instrument) qui seront sauvés sur le disque dur dans le répertoire C:\TPs Instrum\xx où xx est le nom du binôme (ex : A2). Si plusieurs VIs sont demandés, chacun pourra être construit à partir du précédent mais sera sauvegardé séparément sous un nom explicite et non fantaisiste.

***Le travail demandé est dans les zones de texte encadrées et grisées de ce document.***

De plus, un compte-rendu *manuscrit* ( $\neq$  brouillon, 2-3 pages maxi) sera réalisé sur place (sauf mention contraire : TP2). Il contiendra une description très générale du VI, les réponses aux questions posées, et éventuellement une justification des choix réalisés ainsi que les résultats numériques. Des schémas à main levée peuvent être faits mais il est inutile de recopier les fenêtres Labview ! Par contre, n'hésitez pas à écrire des commentaires directement dans les VIs.

Le document manuscrit complète le ou les fichiers VI sauvé(s) sur disque dur et le tout sera évalué en même temps.

*Remarque : il est impossible, pour des raisons pratiques et de licence de travailler sur les VIs en dehors des séances de TPs.*

Il sera tenu compte lors de l'évaluation, de la démarche et de l'autonomie des élèves, du travail d'équipe, et il n'est pas forcément nécessaire de tout réaliser pour obtenir une bonne note.

**Ces séances doivent être considérées comme des examens, c'est à dire que le travail ne doit être fait que pendant le temps imparti. Il sera aussi tenu compte, lors de l'évaluation, des retards et absences non justifiés.**

*Enfin, du matériel informatique et électronique parfois onéreux ou pouvant présenter des dangers (rares !) est mis à disposition : il est vivement recommandé aux élèves de toujours agir en connaissance de cause (une documentation complète peut être mise à leur disposition).*

## **2. Contenu de ce document :**

Le **chapitre 1** contient un **tutorial** Labview qui sera exécuté pendant la première séance, non évaluée.

Le chapitre 2 « l'échantillonnage », contient les principales règles et propriétés de l'échantillonnage, **il est nécessaire de l'avoir assimilé avant d'aborder le premier « vrai » TP** (donc avant la deuxième séance).

Les 3 autres chapitres sont relatifs aux 3 TPs et ne seront donc pas forcément vu dans l'ordre. Il est très fortement conseillé de les avoir lu, et d'en avoir assimilé les généralités, avant d'aborder le TP correspondant.

**En tout état de cause, ne pas se lancer dans la programmation des instruments avant d'avoir lu et compris le document correspondant.**

## II. INTRODUCTION À LABVIEW

### 1. *Introduction*

Labview est un logiciel développé par National Instruments pour l'instrumentation. Il se caractérise par une programmation entièrement graphique.

De très nombreuses fonctions sont disponibles, notamment pour la génération, l'acquisition et le traitement du signal. *Pensez à explorer les différents menus avant de réécrire les fonctions qui existent peut-être déjà*

(Pensez aussi à utiliser les menus « popup » (ou « surgissant ») avec le bouton droit de la souris.

Programmer avec Labview consiste à réaliser des « instruments virtuels » (VI = Virtual Instrument) composés de 2 éléments complémentaires :

- Une **face avant** (initialement grise) reproduisant les différents contrôles (action de l'utilisateur vers l'instrument, donc entrée d'une variable) et les différents indicateurs (visualisation d'une variable).
- Un **diagramme fonctionnel** (initialement blanc) où l'on retrouve les contrôles, les indicateurs ainsi que des opérateurs. Ce diagramme est le programme.

On passe de l'un à l'autre par Ctrl-E ou par le menu *Fenêtres*.

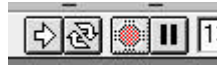
### 2. *Aide*

2 possibilités d'aide en ligne:

- La fenêtre d'aide : menu *Aide – Visualiser l'aide* ou Ctrl-H : si cette aide est active, il suffit de placer le curseur sur un élément pour avoir une aide simplifiée.
- Pour une aide plus complète : menu *Aide – référence en ligne* (et notamment *Labview Référence - Fonctions & VIs*).
- Le jeu de manuels Labview est à votre disposition.

### 3. Exemple :

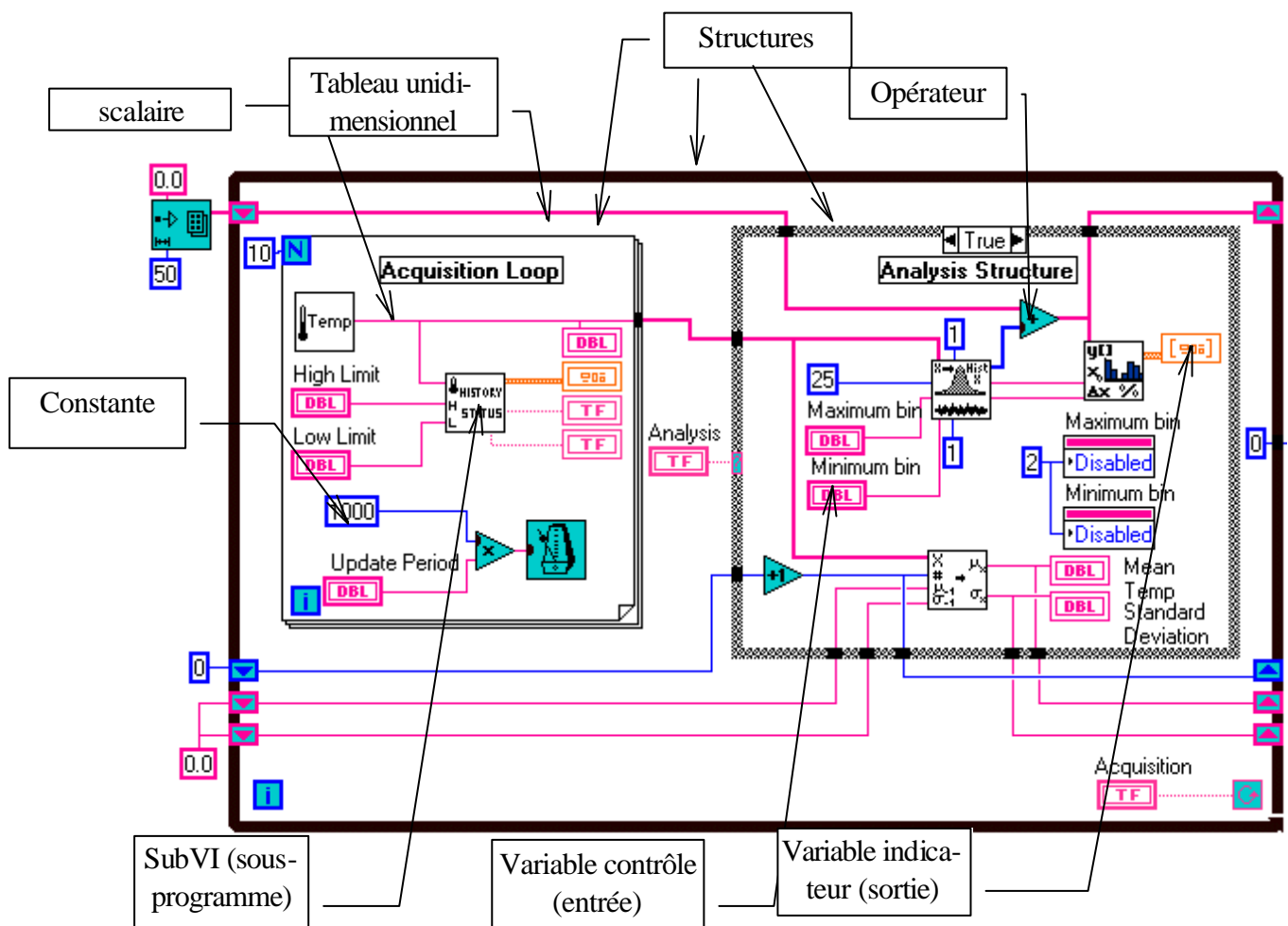
Vous pouvez ouvrir « *EXAMPLES\APPS\TEMPSYS.LLB\temperature System Demo.vi* » pour voir l'aspect d'un VI et de son diagramme, pour l'instant ne cherchez pas à comprendre le fonctionnement.



Lancez l'application à l'aide du premier bouton visualisé ci-dessus. Le deuxième bouton permet de faire tourner une application en boucle et le troisième d'arrêter l'application. Ces 2 derniers ne doivent être utilisés qu'en phase de débogage, il est en effet préférable de gérer le bouclage et l'arrêt.

Faites fonctionner « Température system demo » (qui est une simulation, il n'y a pas d'acquisition...) et expérimentez... On peut faire varier les réglages, les graduations des indicateurs etc...

Visualisez le diagramme fonctionnel :



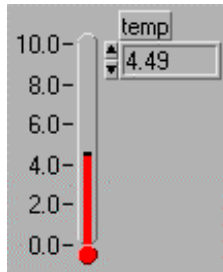
Double-cliquez sur l'icône « History Status », qui est lui-même un instrument virtuel (« subVI »).

La face avant de ce subVI s'ouvre.

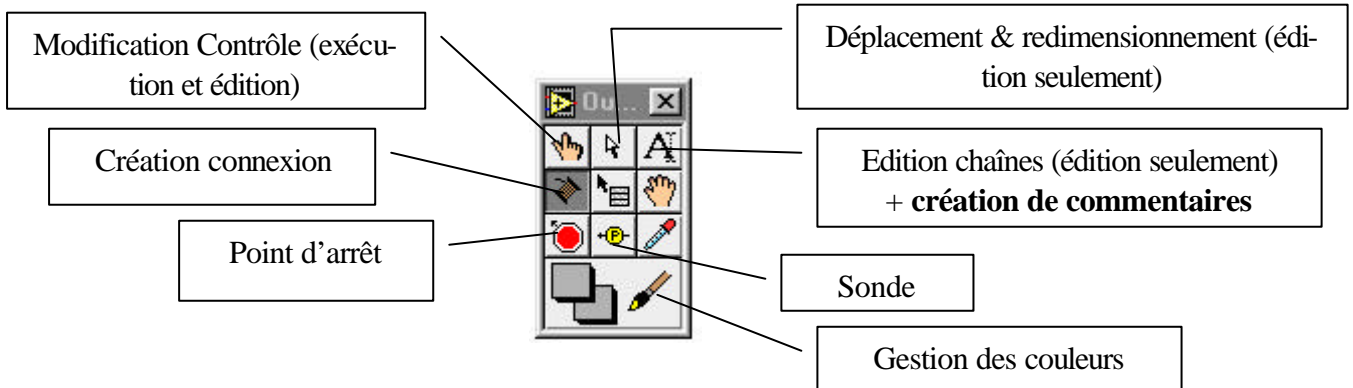
## 4. Construction d'un premier instrument virtuel :



On va réaliser un instrument qui **simule** l'acquisition et l'affichage d'une température.

Ouvrir un nouveau VI (Fichier - Nouveau). Cliquez-droit dans la fenêtre « face avant » (fond gris) pour ouvrir un menu popup « Contrôles ». Choisissez un indicateur de type thermomètre dans la palette « Numérique » (la première), et appelez le « température ». Si la zone « d'étiquette » disparaît avant, cliquez-droit sur l'indicateur et choisissez « Visualiser...étiquette » dans le menu popup.

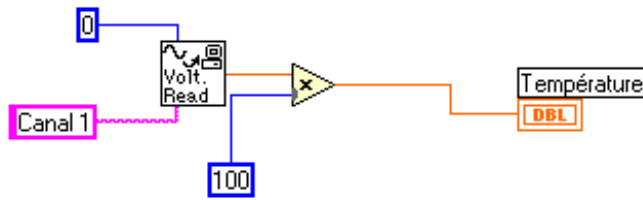



Faire apparaître, si ce n'est déjà fait, la palette d'outils (Fenêtre / Palette d'outils) :



Graduez le thermomètre de 0 à 100 (à l'aide de  ou ).

Basculez sur la fenêtre diagramme fonctionnel (Ctrl-E) (« diagramme », la blanche). Cliquez-droit dans la fenêtre pour faire apparaître « Fonctions ». Sélectionnez « Demo Voltage Read VI » dans la palette « Tutorial », puis l'opérateur multiplication (palette « Numérique »), des constantes numériques (palette Numérique) et une constante chaîne (palette Chaînes de caractères) pour réaliser le diagramme suivant :



Utiliser l’outil câblage  (de la palette d’outils) pour faire les connexions. Il est pratique d’ouvrir la fenêtre Help (Ctrl-H) pendant cette étape pour voir plus précisément les connecteurs des objets que l’on soude : en effet, ils clignotent dans cette fenêtre lorsque l’on passe dessus avec l’outil câblage.

Exécuter avec l’un ou l’autre des boutons  ou .

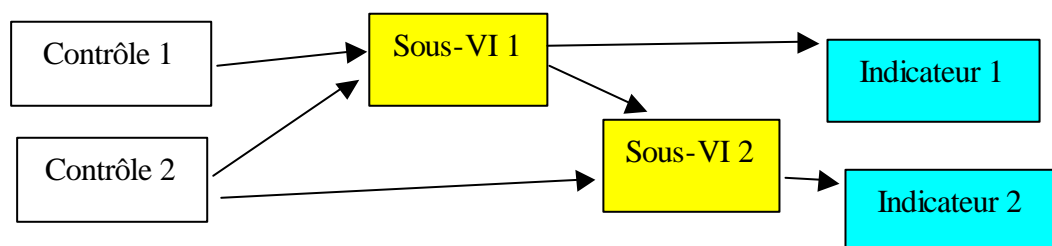
(Attention, il ne s’agit que d’une simulation, pour le moment...)

Notez, à partir de la palette « fonctions » de la fenêtre « diagramme fonctionnel » et de ses sous-palettes, la très grande quantité de fonctions disponibles (plusieurs centaines).

## 5. Principes de Labview

Les objets placés sur la « face avant » sont les variables du programme (A part bien sur les décorations). Ils peuvent être soit des **contrôles**, ou variables d’entrée, qui permettent à l’opérateur de « rentrer » des données, soit des **indicateurs**, ou variables de sorties, qui permettent de visualiser des données.

Le diagramme contient, outre les représentations des contrôles et indicateurs, des opérateurs ou sous-VIs que l’on peut considérer comme les procédures du programme. Ces éléments sont reliés par des connexions qui représentent les flux de données : ces flux sont sortant pour les contrôles, entrants pour les indicateurs et entrant et/ou sortant pour les opérateurs.



La condition nécessaire pour qu’un opérateur soit exécuté (et donc que ses sorties soient disponibles) est que toutes ses entrées soient disponibles. Il en est de même pour les indicateurs. Ainsi, dans l’exemple précédent, l’indicateur « Température » est dépendant de l’opérateur Multiplication, lequel est dépendant de « Volt Read ».


Labview 5 est « multitâches », ce qui signifie que plusieurs groupes d’éléments sans relation de dépendance entre eux seront exécutés « simultanément ».

## 6. Techniques de débogage :

### a. Sondes

Afficher le diagramme pendant l'exécution d'un VI. Faites surgir un menu popup du fil sortant de l'icône du thermomètre, et choisissez "Sonde". La sonde affiche en permanence les données correspondant à la liaison où elle est attachée.

### b. Observation des flux de données

Revenir au diagramme. Cliquer sur le bouton ampoule  pour qu'elle s'allume. Lancer l'exécution. Des points mobiles indiquent le flux des données le long des connections. Egalement, des indications sur les valeurs sortantes apparaissent sur les noeuds actifs.

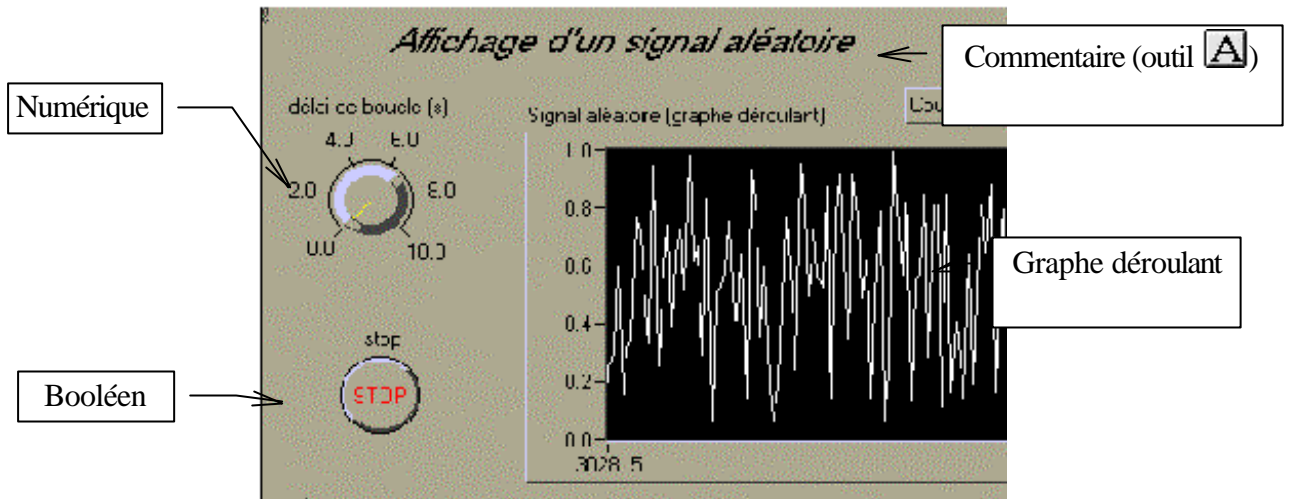
Une expérience de ce type montre bien la logique de la séquentialité dans Labview. Chaque noeud peut s'exécuter dès que toutes les données sont arrivées aux bornes d'entrée. La disposition géométrique des noeuds dans le diagramme n'a aucune influence.

Il existe aussi des possibilités de point d'arrêt.


## 7. Les structures (Palette Structures):

### a. Boucle « While » (tant que)

**Construisons ici un VI qui génère et affiche des données aléatoires. Un sélecteur en face avant permettra d'ajuster la période d'affichage entre 0 et 2 s et un interrupteur permettra d'arrêter.**






Disposer les éléments en face avant suivant le schéma ci-dessus. L'afficheur est du type « *graphe déroulant* » (menu popup *Commandes - Graphes*). C'est donc un affichage point par point, contrairement à « *graphe* » qui affiche tout un tableau d'un seul coup.

Pour cet afficheur, activer *Visualiser / Afficheur numérique* (menu popup) et désactivez *Palette et Légende*. Changez les graduations proposées par défaut par celles indiquées sur la figure (avec )

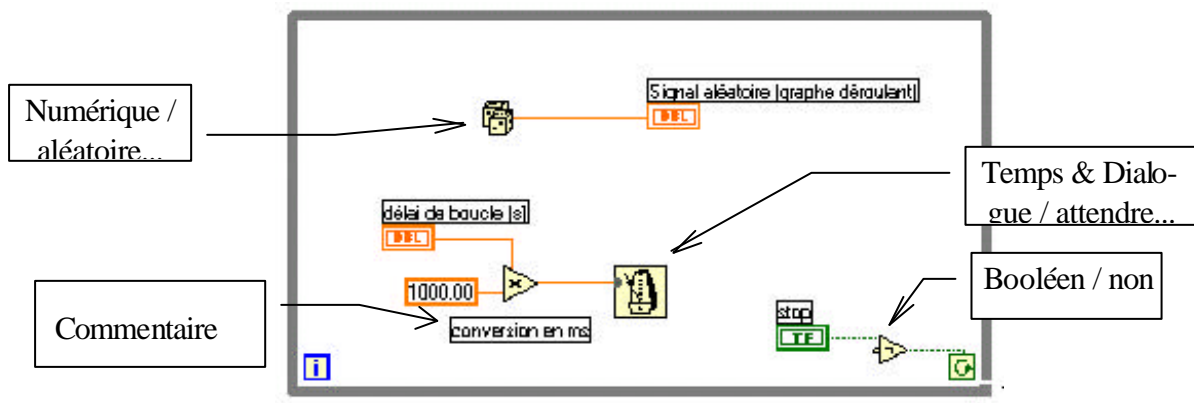




Le diagramme s'inscrit dans une boucle *Tant Que (While Loop)* :


(palette « *structures* », cliquer puis glisser...)

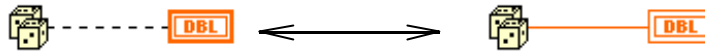
Il y a une borne lecture d'index  et une borne écriture de condition de fin de boucle . La boucle s'exécute tant que le booléen branché sur  est « *true* ».


Le diagramme à réaliser est le suivant :



L'ensemble  —  placé dans une structure fait en sorte que la durée totale d'une itération de cette structure soit *au minimum* de 500 ms.

Il peut arriver qu'il soit nécessaire de changer un contrôle en indicateur ou réciproquement, cela se fait par le menu popup. Remarquer que les contrôles sont représentés par un trait plus gras. Remarquer aussi, dans le premier cas, le trait pointillé indiquant un problème, ainsi que la « flèche brisée » .

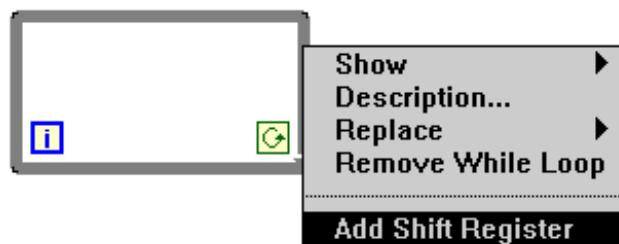






Cliquer sur  pour avoir des explications sur les causes du problème. Les fils « mauvais » (en pointillé) superflus (et parfois bien cachés...) peuvent tous être effacés avec « *Edition / Supprimer les fils incorrects* » ou Ctrl B.

Remarquez que les couleurs correspondent aux types : bleu = entier, orange = réel, violet = string, vert = booléen....

**Nous allons maintenant chercher à afficher, en plus, la valeur moyenne de toutes les valeurs depuis le début :**

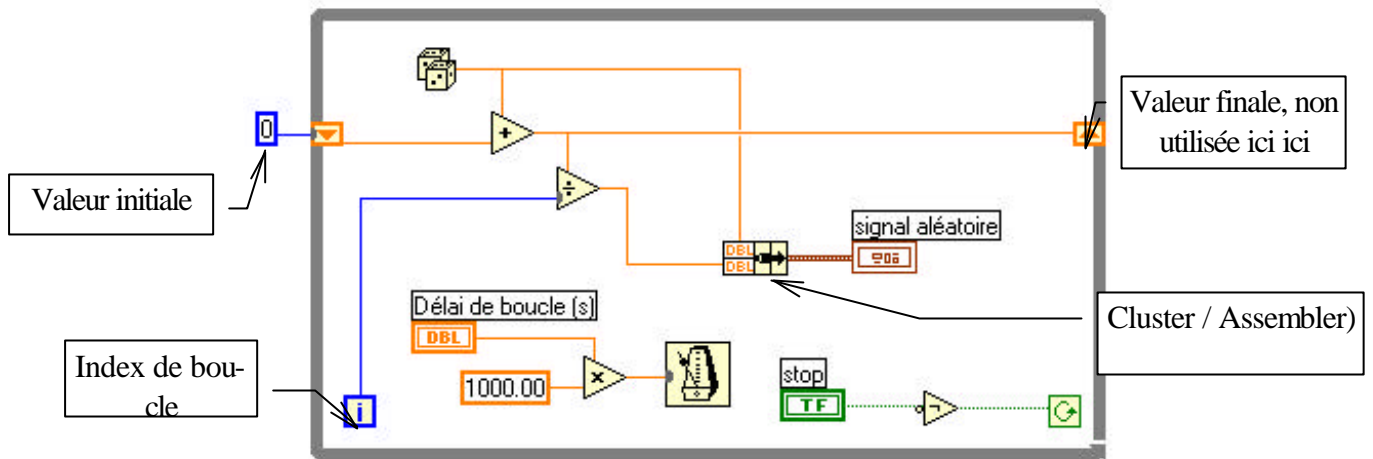
*Il faut donc que dans chaque itération de la boucle on ait accès à la valeur de la somme lors de l'itération précédente. Cela se fait en utilisant un registre à décalage que l'on fait apparaître à partir d'un popup ( clic droit) sur un des bords verticaux de la boucle :*



*Il apparaît les symboles  et  sur ces bords. La borne de droite  est l'entrée où l'on va brancher la valeur de somme actuelle, celle de gauche  est la sortie où la récupère la même valeur mais de l'itération précédente.*

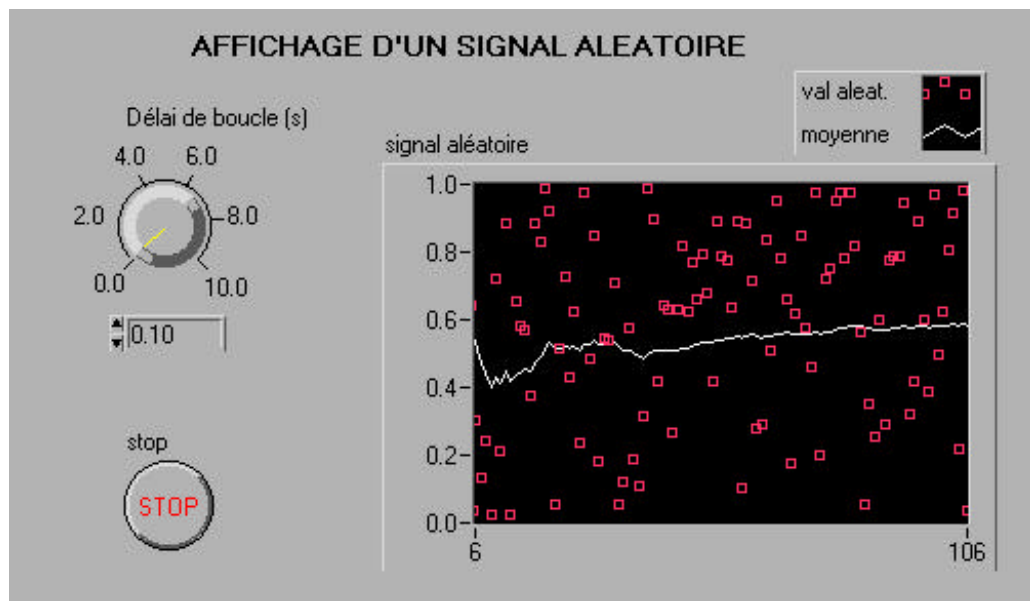
Ce registre doit être initialisé « par la gauche » à l'extérieur de la boucle.

Le diagramme à obtenir est le suivant :



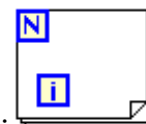
L'afficheur peut accepter plusieurs entrées si elles sont regroupées en « cluster » (ensemble de variables pouvant être de types différents, « enregistrement » du Pascal).

De retour sur la face avant, vous pouvez personnaliser votre afficheur en lui rajoutant différentes options afin d'obtenir quelque chose ressemblant à :



On a affiché ici la « légende » et la « palette » de l'afficheur (menu popup). La légende est redimensionnable à la souris, n'oubliez pas son menu popup.

**b. Boucle « N fois (For Loop) »**



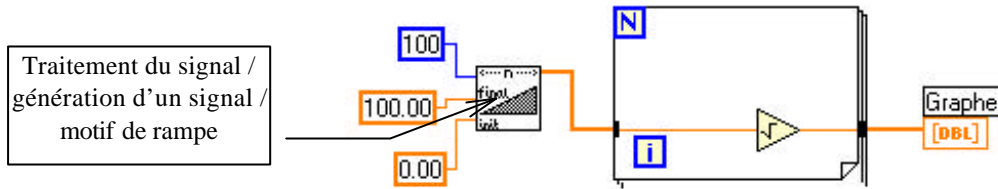
Cette boucle effectue un certain nombre de fois ce qu'elle contient :

**N** est la borne d'entrée où l'on câble le nombre d'itérations désiré,

**i** est la borne de sortie indiquant l'indice de l'itération en cours.

Cette structure accepte aussi la temporisation ainsi que les registres à décalage.

On peut traiter facilement les éléments d'un tableau que l'on câble directement à travers la structure :



*Ce VI calcule et affiche la racine carrée des éléments d'un tableau*

À gauche, un tableau rentre dans la boucle par un « tunnel ». Le trait épais représentatif d'un tableau unidimensionnel s'est transformé en trait fin correspondant à la *i*-ième valeur du tableau. La borne N de la structure n'est pas connectée. Le nombre d'itérations est déterminé automatiquement d'après la taille du tableau. Inversement, quand on connecte un scalaire à l'intérieur de la boucle vers l'extérieur, on récupère automatiquement un tableau de dimension N; dans la figure ci-dessus, cela correspond à la sortie d'une connexion à droite vers un afficheur graphique. Notez que l'on aurait pu faire l'économie du générateur de rampe en connectant l'indice de boucle à la racine carrée.

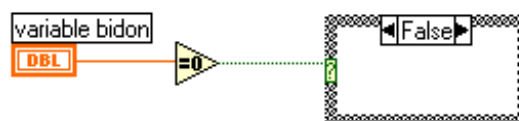
Ce processus (auto-indexing) peut être désactivé par l'option "Désactiver l'indexation" du menu popup du *tunnel* (petit rectangle noir sur le bord de la structure, où les connexions pénètrent).

*Réaliser un VI affichant (à l'aide d'un « grapheXY ») sous forme de courbe paramétrée à 100 points, un cercle. On devra alors construire 2 tableaux (X et Y) et les assembler (cf aide « Ctrl H »).*

*A noter que beaucoup de traitements de tableaux peuvent être faits sans passer par des opérations élément par élément ; par exemple, additionner 2 tableaux de même taille se fait simplement avec un opérateur addition. On peut aussi faire des opérations mixtes scalaire/tableau (par exemple multiplier tous les éléments d'un tableau par une même constante).*

### c. Structure « Condition » (Case ou If then Else)

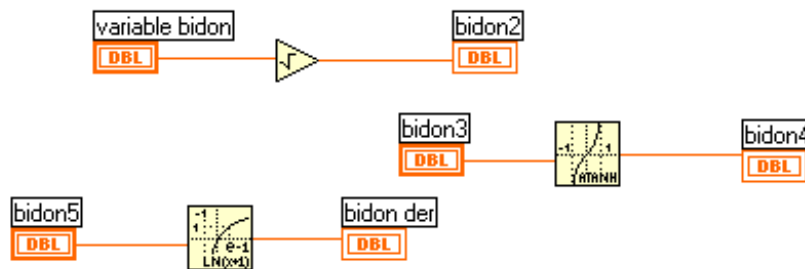
C'est une fenêtre à multifaces à laquelle doit être relié un booléen (issu souvent d'un opérateur comparaison, la fenêtre a alors 2 faces : « true » et « false »), soit un entier (autant de faces que l'on veut, équivalent du Case du Pascal).



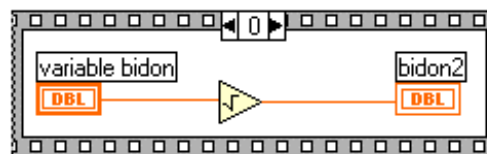
*Application : Réaliser le bornage supérieur, ou écrêtage (limitation à une valeur maxi) puis l'affichage d'une valeur entrée avec un bouton rotatif. Ne pas l'effacer afin de l'utiliser un peu plus loin.*

#### d. Structure Séquence

Le diagramme suivant contient 3 éléments totalement indépendants (et bidons..., ne pas le réaliser). Rien ne permet de savoir dans quel ordre ces 3 éléments seront exécutés, ils seront même probablement exécutés simultanément :



Si cet ordre est important pour le bon déroulement du programme, on doit alors utiliser une structure « séquence » :



On peut rajouter, supprimer ou déplacer des étapes avec un click droit sur un bord de la séquence.

Des variables peuvent être passées d'une étape aux étapes suivantes avec un click droit sur un bord et « Ajouter une variable locale de séquence ».

*Pas d'exemple à réaliser !*

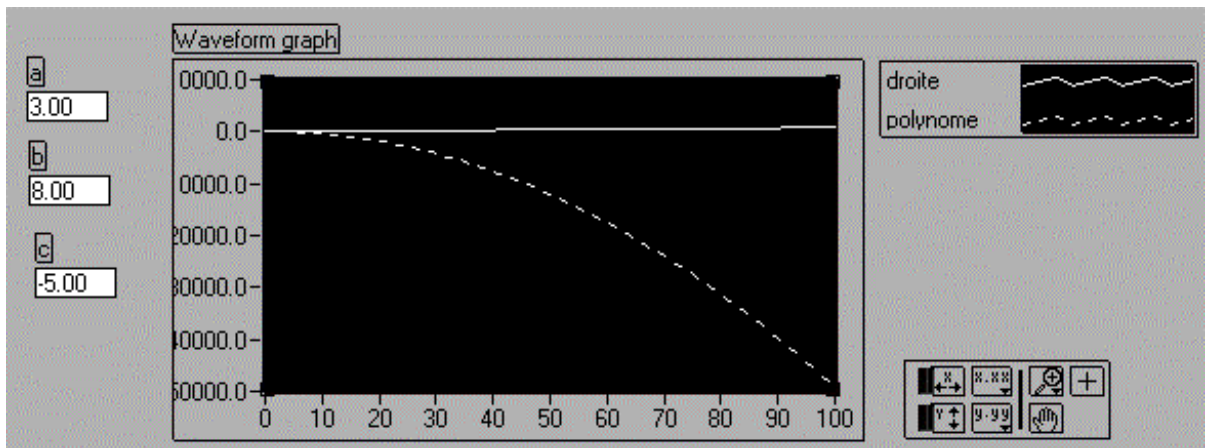
#### e. Structure formule

Il peut être parfois fastidieux de « câbler » une formule complexe. Aussi les concepteurs de Labview ont prévu une exception à la programmation graphique : La structure « Boite de calcul » permet de taper directement une formule.

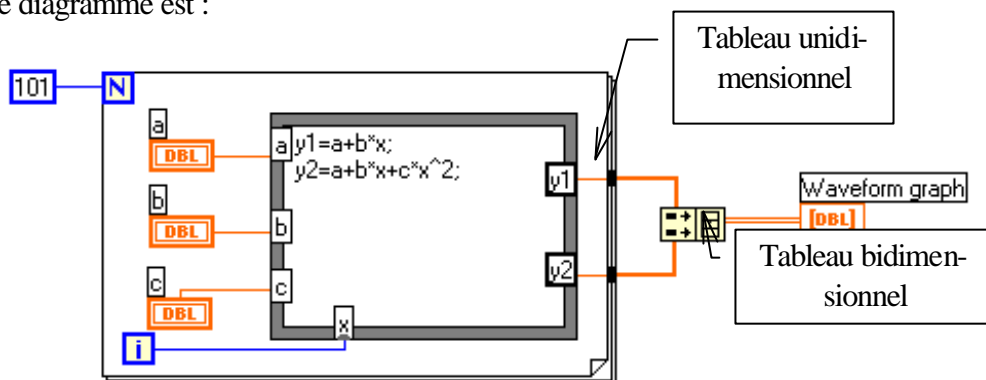
Le menu popup sur le bord de la structure permet de créer les éléments «entrée» et « sortie ». Les textes identificateurs que l'on doit mettre dans ces bornes doivent être repris à l'intérieur pour taper la formule.

Chaque formule doit se terminer par « ; », voir l'aide « Ctrl H » pour connaître les opérateurs et fonctions disponibles.

Vous pouvez réaliser le VI suivant :



dont le diagramme est :




Remarquer que les 2 tableaux unidimensionnels (vecteurs) sont réunis en un tableau bidimensionnel (matrice) grâce à *Build array* (dans *Array & Cluster* redimensionné à la souris).


## 8. Sous VIs

Il est possible de transformer facilement tout VI en un « sous-VI » (fonction avec passage de variables par valeur) utilisable dans un autre VI appelant.

Dans ce cas, tout ou une partie des contrôles seront les variables d'entrée du sous-VI, et les indicateurs deviendront les variables de sortie.

Pour cela, cliquer-droit, à partir de la fenêtre « face-avant », sur l'icône du VI, en haut à droite, sous-menu « Visualiser le connecteur ». Un carré, contenant autant de zone que le VI a de contrôles et d'indicateurs apparaît. Vous pouvez modifier les zones (nombre ou répartition) par le sous-menu popup « Modèles ».

Ensuite, toujours dans la fenêtre « face avant », il faut connecter les zones avec les contrôles et indicateurs à l'aide de l'outil  , veillez à ce que les contrôles soient à gauche et les indicateurs à droite.

 Le sous-VI peut être utilisé à partir de la fenêtre diagramme du VI appelant grâce au bouton du menu popup “Fonctions” qui permet de charger tout fichier VI.

Exemple: Réaliser un sous VI “écrétage” à l'aide du VI réalisé plus haut (3 entrées et 1 sortie).

**Important** : par défaut, Labview ne sauve pas les valeurs des contrôles, même si elles ont été modifiées. Pour conserver de nouvelles valeurs (qui deviendront celles par défaut), cliquer sur le menu « *Execution / Prendre les valeurs actuelles par défaut* » puis enregistrer le VI.

## 9. Conclusion :

Vous connaissez maintenant environ 10% des possibilités de Labview, alors explorez, déroulez les menus, surgissez les popup !

Pour approfondir :

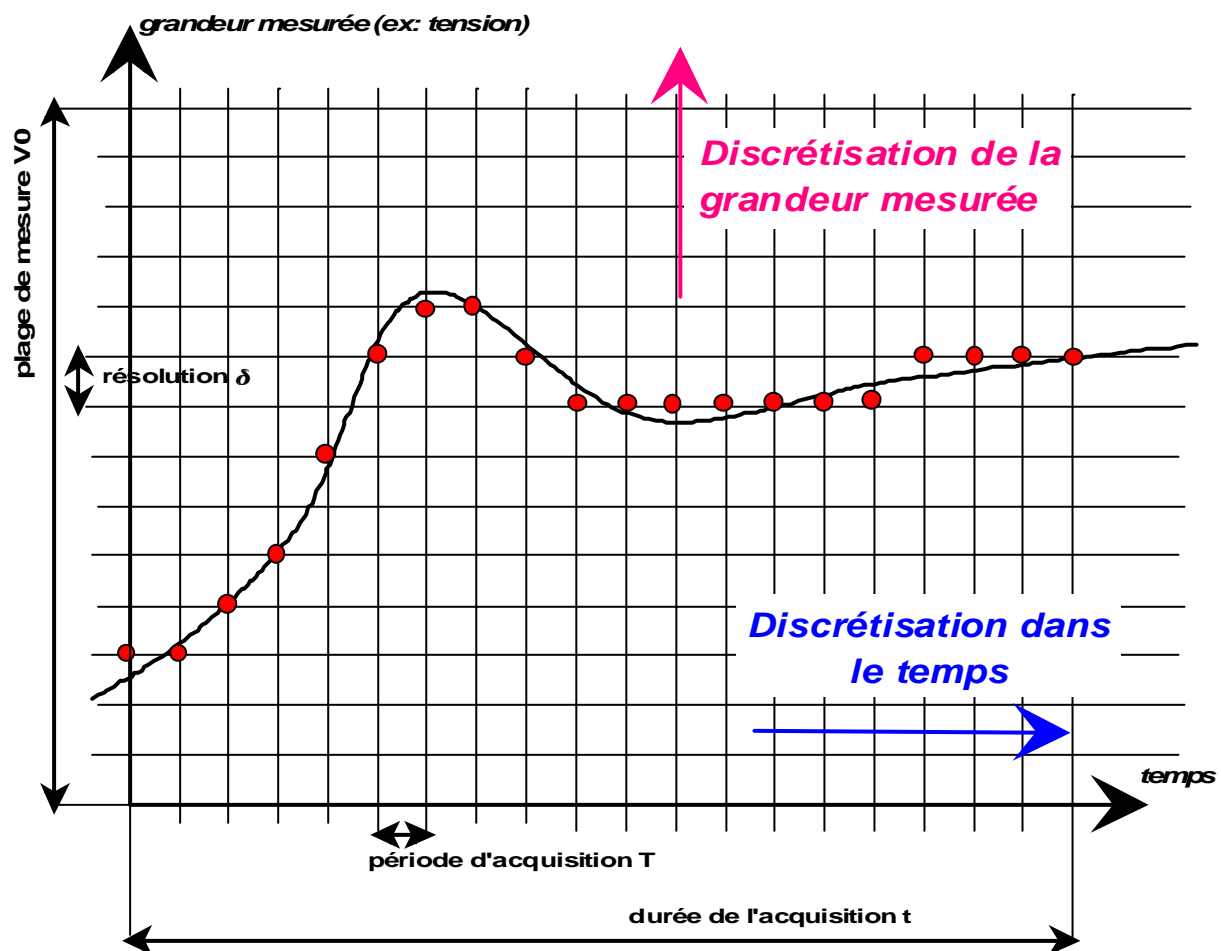
- livre « tutorial », en Français ,
- Les excellents exemples de ni\Labview\exemples\
- Les manuels de référence.

### III. L'ECHANTILLONNAGE

L'échantillonnage est l'opération qui consiste à faire **l'acquisition d'un signal** à l'aide d'un **système numérique**.

Soit un signal continu réel (tension par exemple)  $V(t)$  : l'échantillonnage va consister à **numériser périodiquement la valeur de ce signal**. On effectue alors une **double discrétisation** de ce signal :

- **Discrétisation dans le temps** : le signal ne sera connu que toutes les  $T$  s,  $T$  étant la période d'échantillonnage, et le nombre de points acquis sera fini (pour une durée d'acquisition finie),
- **Discrétisation de la grandeur mesurée** : En numérique, une valeur correspond toujours à un nombre fini de bits (ou d'octets...), d'autre part la « plage de mesure » d'un dispositif d'échantillonnage est finie, par conséquent, le nombre de valeurs mesurables différentes est aussi fini.



Ces discrétisations sont donc des **sources d'erreurs** :

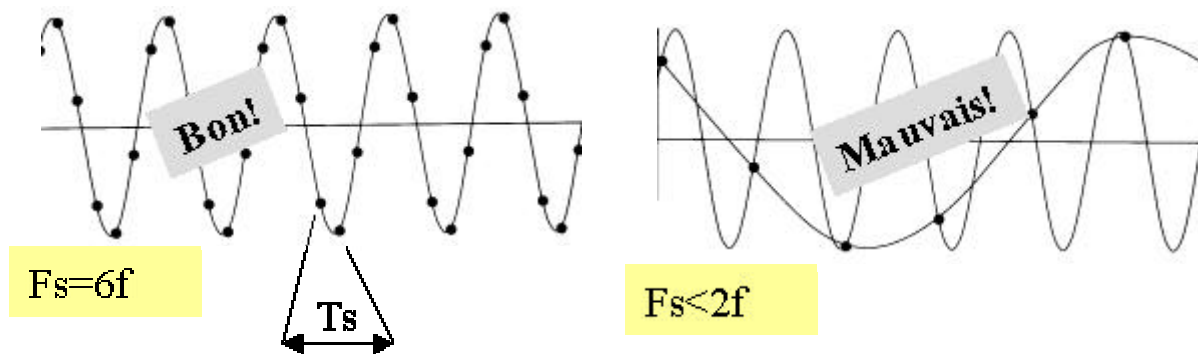
## 1. La discrétisation dans le temps :

L'utilisateur peut généralement choisir la **fréquence d'échantillonnage** (inverse de la période), jusqu'à une limite supérieure, fonction du matériel utilisé (TPs 1 et 2 : carte d'acquisition, 200 kHz, TP3 : module d'acquisition, une acquisition toutes les 60 ms).

Si le signal échantillonné est **rapidement variable**, son échantillonnage peut alors poser quelques problèmes. Un échantillonnage sera dit **correct** si l'on peut, à partir des seuls points échantillonnés, **reconstituer le signal** (par exemple par interpolation entre les points). Supposons que ce soit un signal sinusoïdal de fréquence  $f$ : Le théorème fondamental de l'échantillonnage, dit « **théorème de Shannon** », nous dit que :

**L'échantillonnage d'un signal de fréquence  $f$  doit se faire avec une fréquence d'échantillonnage  $F_s$  supérieure à  $2f$ .**

Le non suivi de cette règle essentielle entraîne pratiquement à coup sûr des résultats aberrants.

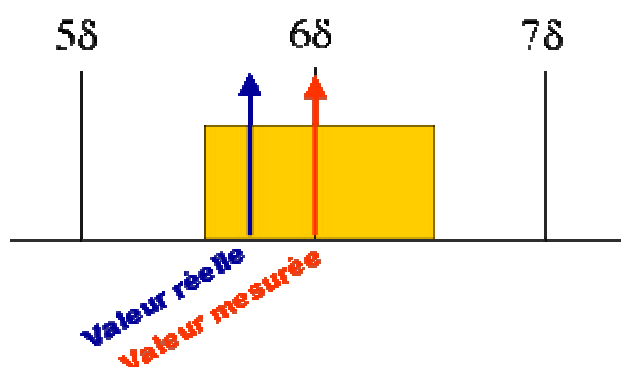


L'illustration et la justification pratique de ce théorème seront vues durant le TP2, la démonstration sera vue pendant le cours d'UP1

## 2. La discrétisation de la grandeur mesurée :

La discrétisation intervient lorsque les valeurs finales constituent un **ensemble discret**. On peut alors définir un pas de mesure  $\delta$  (ou parfois digit, ou LSB, Least Significant Byte) qui est le plus petit écart mesurable.

C'est la **résolution** du système d'acquisition, à ne pas confondre avec *l'incertitude*, ou la *sensibilité*.



On introduit alors une erreur aléatoire, dite erreur de discrétisation.

On montre que son écart-type est :  $\mathbf{S}_d = \frac{d}{2\sqrt{3}}$

### 3. *Le théorème de la moyenne :*

De plus en plus souvent, le système d'acquisition utilisé a des performances permettant d'avoir une **fréquence d'acquisition grande** par rapport à la cadence d'acquisition nécessaire pour une application donnée. Ainsi, pour le TP1 (régulation de température), 10 mesures par seconde suffisent alors que le système est capable d'échantillonner à 200 kHz !

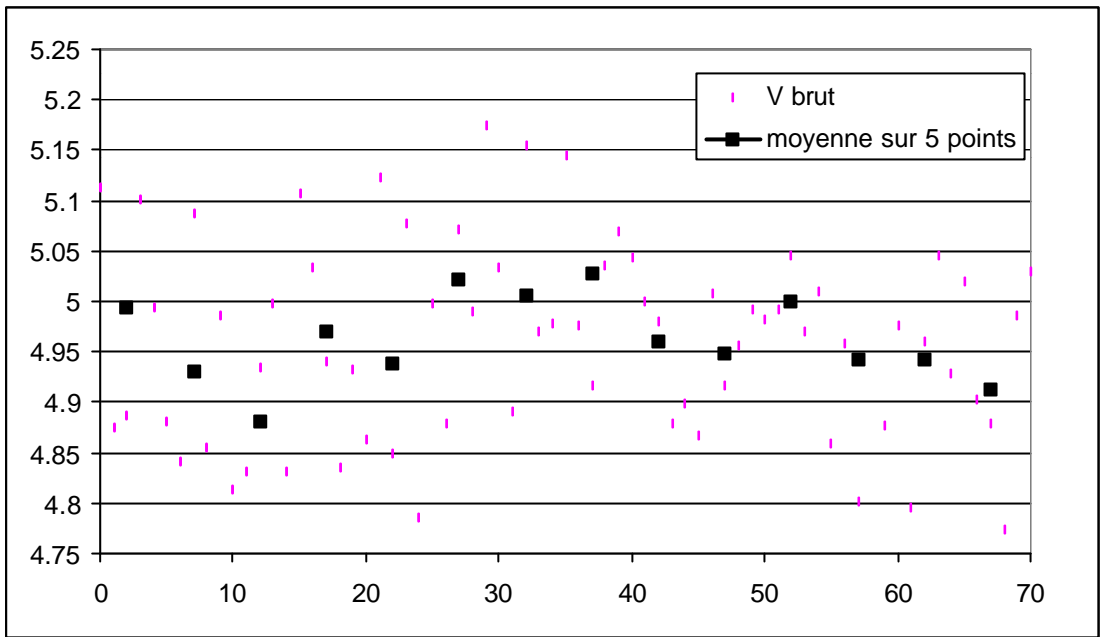
On peut mettre à profit ces capacités en **moyennant un grand nombre de mesures**, en effet, le théorème de la moyenne nous indique que :

- Si l'on prend comme **nouvelle mesure la moyenne** de n mesures élémentaires consécutives :  $\bar{x} = \frac{1}{n} \sum_1^n x_i$
- Si ces n mesures ont une composante d'erreur **aléatoire** et **indépendante** d'écart-type  $\sigma$ ,

Alors l'écart-type de la contribution de cette erreur sur la moyenne s'écrit :

$$\mathbf{S}(\bar{x}) = \frac{\mathbf{S}(x)}{\sqrt{n}}$$

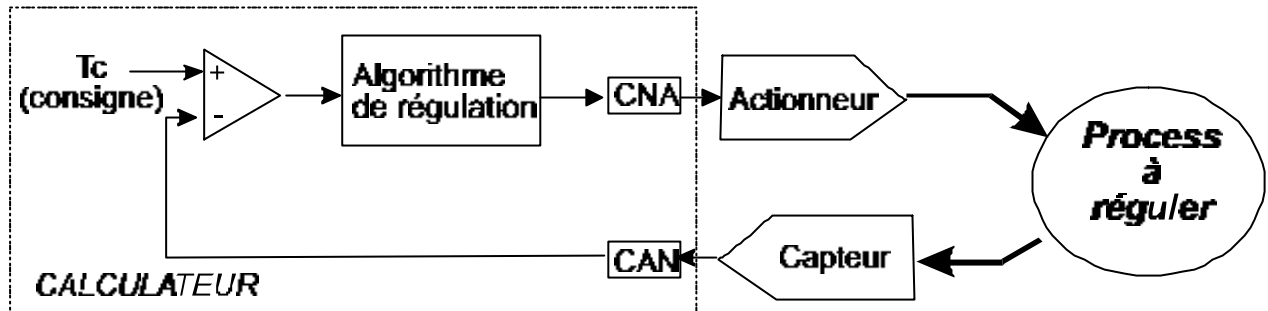
Ceci permet de réduire considérablement la composante aléatoire et indépendante de l'erreur (appelée parfois « bruit »), mais est bien sûr sans effet sur les autres sources d'erreur (erreur d'étalonnage, dérives lentes etc...). De plus, l'acquisition dure n fois plus longtemps.





# IV. TP N°1 REGULATION NUMERIQUE DE TEMPERATURE

## 1. Principe général:



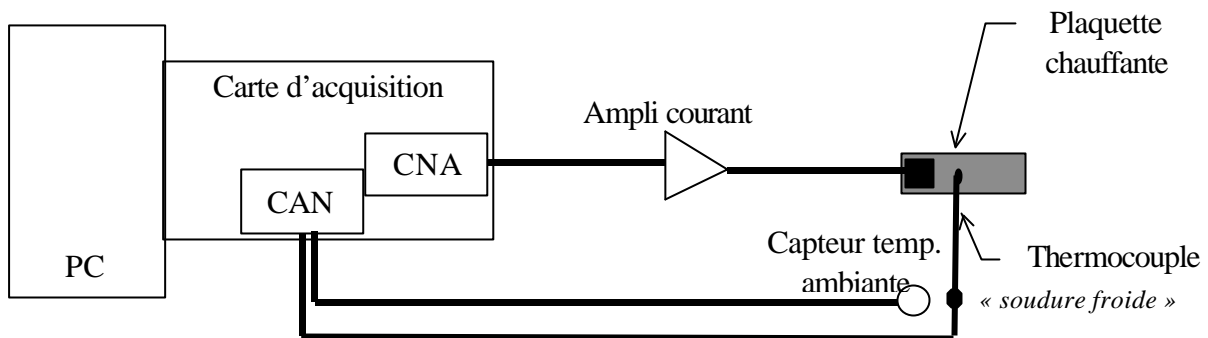
Le but de la régulation de température est de maintenir la température  $T(t)$  d'un système la plus proche possible d'une température  $T_c(t)$  dite *de consigne* (choisie par l'utilisateur et pouvant être variable), et dans de bonnes conditions de stabilité. La régulation se fait en 3 étapes:

- 1) - Mesure de la température  $T(t)$  à l'aide d'un Convertisseur Analogique – Numérique (CAN).
- 2) - Calcul de l'action  $P(t)$  à appliquer au processus (= puissance de chauffage) par un algorithme approprié .
- 3) - Application de cette action à l'aide d'un Convertisseur Numérique – Analogique (CNA).

Ce cycle est appliqué périodiquement avec une période dite *période de régulation*, inférieure à la seconde dans ce cas.

Les notions vues au cours de ce TP seront acquises plus en détail lors du cours d'UP3, on pourra voir aussi : <http://www.emse.fr/~pbreuil/auto/index.htm>

## 2. Dispositif, action et mesure :

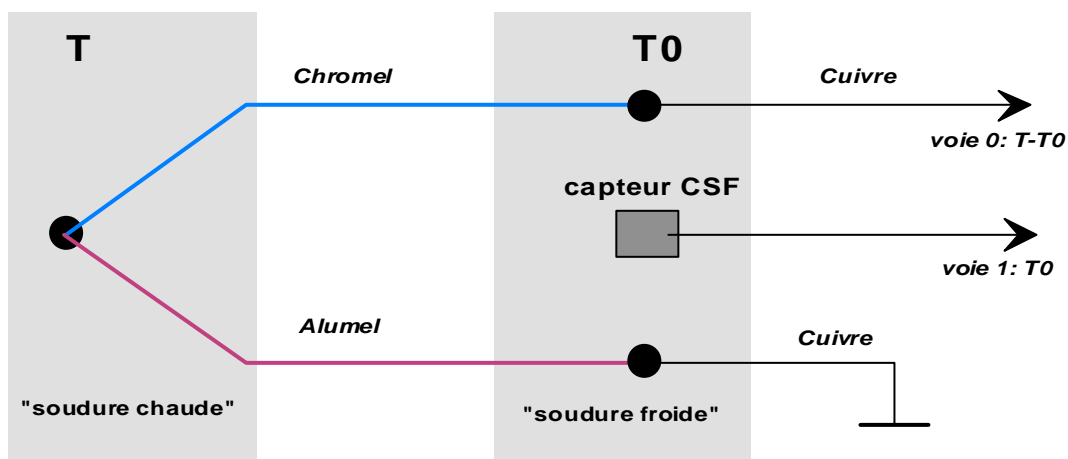


On dispose d'une micro-plaquette chauffante (partie mesure : maxi environ 200°C, attention, l'élément chauffant peut atteindre 800°C !) dont on peut mesurer la température à l'aide d'un thermocouple.

La plaquette est chauffée par une source de tension issue de la voie 0, amplifiée en courant, du **convertisseur numérique analogique** 0 –10 V. Il est donc possible de piloter la puissance de chauffage à l'aide de Labview.

On utilise pour la mesure de température un **thermocouple** de type K (Chromel – Alumel) collé près de la source de chaleur. Un thermocouple est un capteur fournissant une tension sensiblement proportionnelle à la **différence de température** entre le point chaud (« soudure chaude ») et la soudure avec le reste du circuit (« soudure froide »). L'intérêt du thermocouple est qu'il permet de mesurer jusqu'à plus de 1000 °C.

Mais il ne mesure qu'une **différence de température**, il est donc nécessaire de connaître la **température T0 de la soudure froide**, proche de l'ambiante, à l'aide d'un autre capteur (dit CSF « Compensation de Soudure Froide »), de type LM35, fournissant une tension proportionnelle à la température T0, mais ne fonctionnant qu'en dessous de 100 °C.



Le signal du thermocouple (type « K », soit environ 40µV/°C) est envoyé sur la voie 0 du **convertisseur analogique numérique** (carte 6024 de National Instruments).

DE même, la voie 1 reçoit le signal « de soudure froide » issu d'un capteur semi-conducteur (dit de type « IC », il s'agit ici d'un LM35 délivrant 10 mV/°C).

Réaliser un VI permettant d'émettre une action provenant d'un contrôle et de visualiser l'évolution de la température. On essaiera d'améliorer le rapport Signal/Bruit de la mesure tout en veillant à ce que celle-ci reste de courte durée (< 100 ms). Pour cela, on exploitera le théorème de la moyenne (p 19).

On veillera aussi, et ceci est valable pour les exercices suivant, à ce que l'arrêt du programme par un bouton adéquat provoque l'arrêt du chauffage.


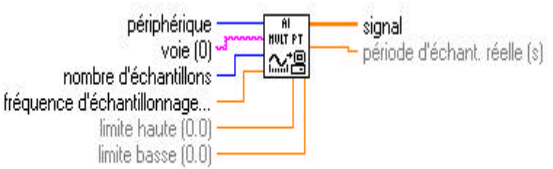

Les outils nécessaires à l'acquisition et à l'action sont dans les palettes « Acquisition de données / Entrées analogiques, Sorties analogiques, Conditionnement de signaux » :

**a. Outils d'acquisition avec carte National Instruments (sous-menu Acquisition de données/ Entrées analogiques) :**

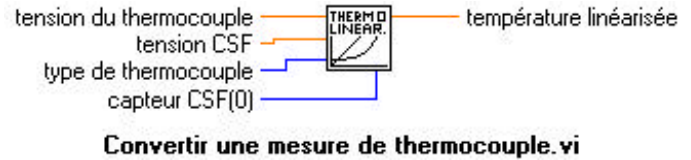
⇒ **périphérique** = numéro de la carte (il peut y en avoir plusieurs), 1 par défaut.

**Voie** = numéro de voie : de 0 à 7, c'est une chaîne de caractère contenant le numéro de voie.

Remarque : la carte d'acquisition contient un amplificateur à gain réglable, le choix du gain se fait automatiquement à partir des 2 entrées « **limite haute** » et « **limite basse** » des outils d'entrée analogique. Ces limites peuvent être calculées à partir des plages de températures et des caractéristiques des capteurs, ces données étant citées plus haut.

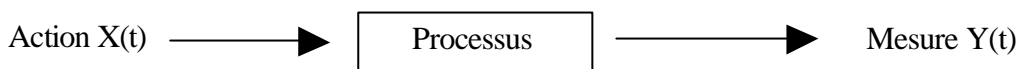
<p>Acquisition <b>d'une tension</b> (Méthode la plus simple)  (AI = Analog Input)</p>	 <p><b>AI-Echantillonner une voie.vi</b></p>
<p>Acquisition <b>tableau de N tensions</b>, à une fréquence déterminée (maxi 200 kHz).  Le calcul de la moyenne du tableau doit permettre de diminuer le bruit</p>	 <p><b>AI-Acquérir un signal.vi</b></p>
<p>Génération d'une tension, la tension est maintenue jusqu'à la prochaine utilisation (AO = Analog Output)</p>	 <p><b>AO-Mettre à jour une voie.vi</b></p>

Calcul température à partir des signaux du thermocouple et du CSF. Cliquer avec le bouton droit sur les connecteurs de type de capteur et sélectionner « créer une constante » afin de pouvoir choisir le type.



Attention, les 2 acquisitions (2 capteurs) ne doivent pas être simultanées.

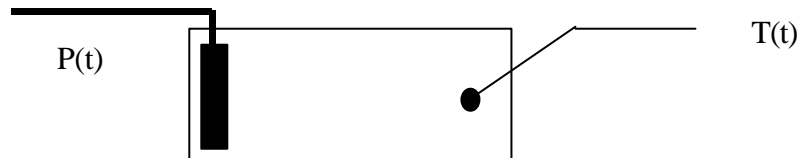
### 3. Identification du processus :



La fonction de transfert d'un processus permet de calculer la mesure  $Y(t)$  en fonction de l'action  $X(t)$ . Son étude théorique fait appel à la transformée de Laplace pour les systèmes continus et la transformée en Z pour les systèmes échantillonnés (qui seront vu dans le cadre de l'UP2).

L'identification est une **modélisation de comportement** du processus qui permet d'écrire la fonction de transfert et de **régler facilement les algorithmes de régulation**. Elle se fait en 2 étapes :

1 : Détermination du type de loi physique à laquelle obéit le processus : Dans notre cas, un processus de type « **Ordre 1 plus retard** » (système obéissant à une équation différentielle d'ordre 1, suivi d'un retard pur, cas le plus courant) est une bonne approximation :



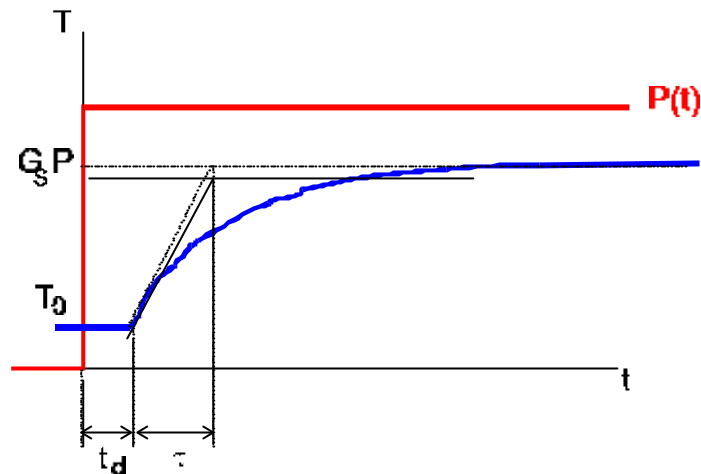
$$CdT = P(t)dt - \frac{1}{G_s}(T - T_0)dt \quad , \quad \text{avec un temps de retard } t_r$$

( $C$  = capacité calorifique,  $P$ =puissance,  $T_0$ = $T$  ambiante)

Le retard  $t_r$  est fonction de la conductivité thermique, de la capacité calorifique et de la géométrie.

La réponse à une action **échelon** ( $P(t)=0$  si  $t<0$  et  $P(t)=P$  si  $t>0$ ) est dans ce cas:

$$T(t) = G_s P + (T_0 - G_s P) \exp\left(-\frac{t - t_r}{\tau}\right) \quad (1) \quad \text{avec } \tau = C G_s$$



2 : **Détermination des constantes du modèle** à partir de la réponse à un échelon d'action. Par soucis de simplification, on considèrera par la suite que l'action est la tension appliquée  $V$  (au lieu de la puissance  $P$ ).

Les 3 paramètres à évaluer sont donc :

- $G_s$ , **gain statique** car  $T=G_s V$  lorsque le régime permanent est atteint (On suppose que la loi  $T=f(V)$  est linéaire, ce qui est généralement une grossière approximation, et qu'il n'y a pas de perturbations, en fait, le gain statique dépendra un peu de la température).
- $t$ , **constante de temps** du système : Si  $C$  est la capacité calorifique, alors  $\tau=CG_s$ .  $\tau$  peut être déterminée à partir de la tangente au point de départ, si l'on connaît le gain statique, ce qui n'est pas toujours évident.
- $t_r$ , **retard pur** du système : il dépend de la conductivité thermique des matériaux et de la géométrie du système, notamment de la distance entre la source de chauffage et le capteur. De plus, on peut considérer, en première approximation, **que la période de régulation,  $Dt$ , s'ajoute au retard.**

a. **Travail à effectuer :**

*On considèrera que la plaquette est un « ordre 1 avec retard » : Déterminer les 3 paramètres  $G_s$ ,  $t_r$  et  $t$  à l'aide d'un VI permettant de visualiser la réponse à un échelon d'environ 50% de la puissance maxi. On soignera tout particulièrement le retard qui, bien que faible, a une influence capitale sur la stabilité de la régulation. Ces paramètres seront utilisés plus loin.*

2 méthodes sont possibles :

- **Méthode graphique classique** (et dépassée ?) : voir schéma ci dessus. Cette méthode nécessite de connaître l'asymptote et peu donc être longue : il faut en effet enregistrer la température pendant au moins 3 ou 4 fois la constante de temps.
- **Méthode de moindres carrés** (« fittage de courbe ») : à l'aide du solveur d'Excel :

Il est alors nécessaire d'enregistrer les couples (temps, température) dans un fichier lisible par Excel (facile avec « E/S fichiers / écrire dans un fichier tableur »).

Sous Excel, ajouter, en plus des 2 colonnes venant du fichier (temps et température mesurée), une colonne de température calculée selon le formule (1) (attention à la gestion du retard !) et faisant appel aux 3 paramètres à déterminer, initialisés arbitrairement dans 3 cellules. Calculer dans une autre cellule la somme des carrés des différences (mesure – calcul).

**Enfin à l'aide du solveur**, minimiser cette dernière cellule en prenant comme cellules variables les 3 cellules paramètres.

La deuxième méthode est beaucoup plus précise et il n'est plus nécessaire d'attendre le régime permanent. Elle est de plus généralisable à d'autres types de processus.

## 4. La régulation PID :

Afin de simplifier, **on prendra comme action la tension de chauffage**  $V(t)$  au lieu de la puissance. On pourra discuter de la pertinence de cette approximation.

La régulation PID est l'algorithme de régulation le plus utilisé. L'action PID est en fait la somme de 3 actions :

### a. L'action Proportionnelle ("P"):

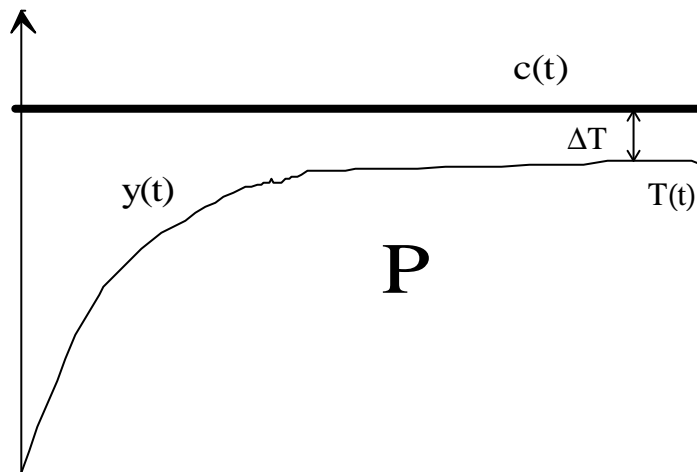
**L'action P est proportionnelle à l'écart  $e(t)$  entre la consigne et le signal:**

$$V(t) = K_p(T_c(t) - T(t)) = K * e(t)$$

Si le « coefficient proportionnel »  $K_p$  (paramètre réglable) est suffisamment faible, le système est stable, **mais la température ne peut alors pas atteindre la consigne** (sinon, on aurait  $V(t)=0$  !): On a alors un écart, dit écart de statisme:

$$\Delta T = e(t) = V(t) / K_p$$

Si  $K_p$  est grand et si il existe un retard, le système se met à osciller d'où un choix dramatique entre la stabilité et la précision...



Réponse à un échelon de consigne

Remarque : Ici, **l'action est bornée** (entre 0 et +10V), il s'agit d'une non-linéarité par rapport à la théorie du PID. Dans le cas plus général,  $K_p$  grand peut entraîner une divergence dramatique.

On montre que celle-ci a lieu si  $KG_s \geq \frac{p}{2} \frac{t}{t_R} \approx 1.57 \frac{t}{t_R}$

Pour avoir en même temps une bonne précision et une bonne stabilité, on doit ajouter:

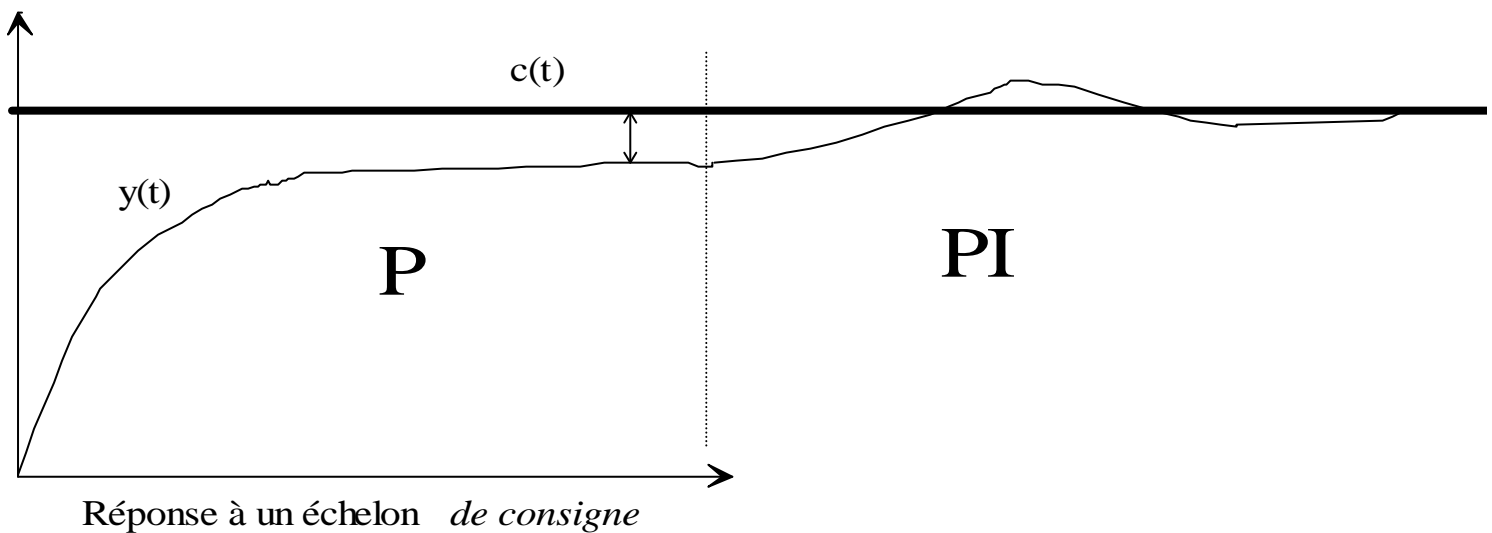
**b. L'action Intégrale ("I")**

Pour compenser petit à petit l'écart de statisme, on ajoute à l'action un terme d'intégration de l'écart consigne - signal .

On a alors:

$$V(t) = Kp \left[ e(t) + \frac{1}{t_i} \int e(t) dt \right]$$

L'action I est caractérisée par le paramètre réglable  $t_i$ , homogène à un temps, est sera d'autant plus forte que  $t_i$  sera faible.



Pour corriger les variations brusques de température (ou de consigne), on peut ajouter encore:

**c. L'action Dérivée ("D")**

On ajoute à l'action un terme de dérivée de l'écart consigne - signal . L'action D est caractérisée par le paramètre réglable  $t_d$ , homogène à un temps, est sera d'autant plus forte que  $t_d$  sera fort.

On a alors l'algorithme PID complet :

$$V(t) = Kp * \left( e(t) + \frac{1}{t_i} * \int e(t) dt + t_d * \frac{d(e(t))}{dt} \right)$$

Soit, en termes « discrets » :

$$V_k = Kp \left( E_k + \frac{t_d}{\Delta t} (E_k - E_{k-1}) + \frac{\Delta t}{t_i} \sum_{j=0}^k E_j \right), \text{ où } \Delta t \text{ est la période de régulation.}$$

*(Par convention, le terme proportionnel  $K_p$  agit sur les 2 autres termes).*

Il existe de nombreuses « recettes » permettant de déterminer les bonnes valeurs des paramètres PID, les principales étant :

■ **D'après les paramètres d'identification du processus** : (nécessité de faire une identification, pas toujours évidente) qui sont, pour un ordre 1 avec retard :

- La constante de temps  $\tau$  (ou « inertie »),
- Le retard pur  $t_r$ ,
- Le gain statique  $G_s$  (rapport température / action en régime « permanent »).

On peut alors estimer les valeurs permettant de réguler en respectant les critères dits de « **Ziegler & Nichols** »: « *Système oscillant amorti d'un facteur 4 entre les 2 premières oscillations* »

$$Kp = \frac{1.2t}{G_s tr} \quad t_i = 2t_r \quad t_d = 0.4 t_r$$

On pourra aussi essayer les critères dits « **CHR0** » :

$$Kp = \frac{0.6t}{G_s tr} \quad t_i = t \quad t_d = 0.5 t_r$$

■ D'après la méthode dites « **des oscillations limites** » (moins efficace que la précédente mais souvent plus facile à mettre en œuvre, car ne nécessite pas d'identification):

*Régler le PID en proportionnel pur, puis augmenter progressivement  $K_p$  jusqu'à ce que le signal commence à osciller, on notera alors  $K_{pl}$  la valeur de  $K_p$  et  $t_0$  la période des oscillations : Les paramètres selon les critères de Ziegler et Nichols seront alors :*

$$Kp = 0.62 K_{pl} \quad t_i = 0.5 * t_0 \quad t_d = t_i / 5$$

*Ces 2 méthodes, plus ou moins empiriques, (qui donnent d'ailleurs parfois des résultats différents) sont très approximatives et doivent être affinées au cas par cas.*

Réaliser un VI de régulation PID. On pourra régler, à partir de la face avant : la consigne, les 3 paramètres de régulation PID ainsi que la période de régulation.

*Pendant le fonctionnement, on réglera les paramètres de régulation à partir d'une des 2 méthodes, puis on « affinera » empiriquement afin d'avoir une régulation :*

*1 : précise*

*2 : rapide (consigne atteinte rapidement)*

*3 : répondant « efficacement » aux changements de consignes ou aux perturbations (courants d'air...).*

*Ces 3 critères sont-ils compatibles ?*

*On étudiera succinctement l'effet de la période de régulation ainsi que celui du rapport signal/bruit de la mesure.*

*On pourra enfin améliorer la régulation en limitant la somme de l'intégrateur.*



## V. TP N°2: DETECTION SYNCHRONE

### 1. **But:**

A : Utilisation d'une carte d'acquisition pour l'acquisition d'un signal (oscilloscope) + analyse fréquentielle.

B : Application à la détection et à la mesure d'un signal périodique de référence connue, ce signal pouvant être noyé dans un bruit (= signaux de fréquences différentes) qui lui est supérieur de plusieurs ordres de grandeur. Cette détection peut se faire de différentes façons.

Ici, il s'agira de détecter, à l'aide d'une photodiode, l'éclairage d'une diode électroluminescente alimentée par un courant sinusoïdal.

On utilise une carte d'acquisition National Instruments 6024E pouvant faire jusqu'à 200000 mesures par seconde.

*Pour ce TP, il n'est pas utile de rendre un compte-rendu écrit, les remarques et réponses pourront être incluses dans le VI sous forme de commentaires.*

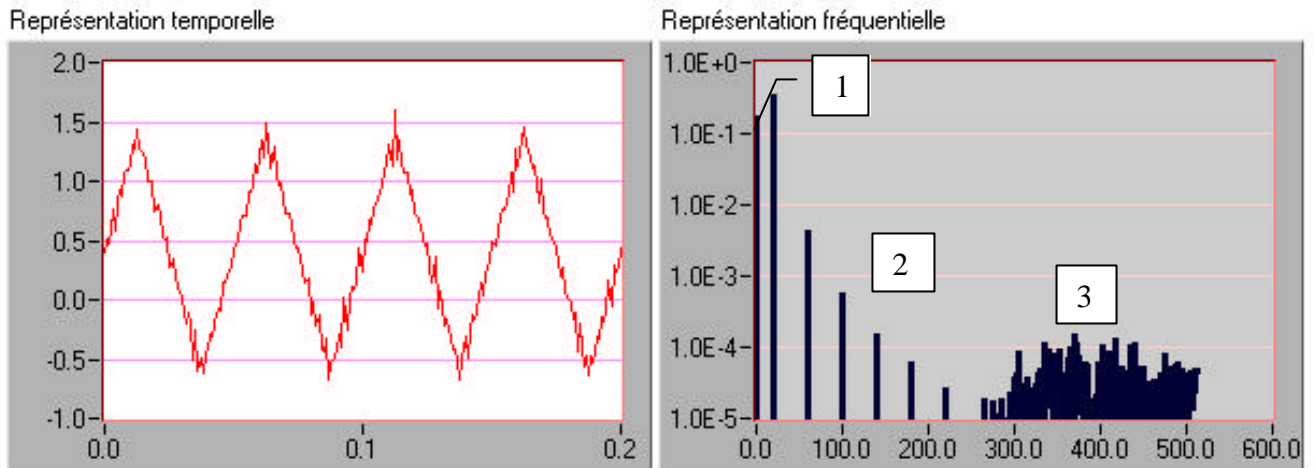
### 2. **Préambule : Analyse fréquentielle & signaux périodiques**

(Ces notions, déjà connues par certains, seront précisées lors du cours d'UP1)

Pour la suite, nous nommerons « signal » toute série, ou **tableau unidimensionnel**, de mesures acquises avec un intervalle de mesure  $T$  dont l'inverse est la fréquence d'acquisition  $\tau$  (ou d'échantillonnage).

L'information contenue dans un signal  $f$  peut être visualisée de 2 façons différentes :

- **Dans le domaine temporel** : méthode classique, fonction  $f_k = x(kT)$
- **Dans le domaine fréquentiel** : permet de représenter la répartition de l'intensité (ou de la puissance) du signal dans les différentes fréquences du spectre :



- 1 : composante continue (= valeur moyenne)
- 2 : Signal triangulaire (1 fondamental + des harmoniques)
- 3 : « bruit » Gaussien haute fréquence

Le passage du domaine temporel vers le fréquentiel se fait par la **transformée de Fourier**. Comme le signal est **échantillonné** (on ne connaît sa valeur que  $f_s$  fois par seconde), on utilise la transformée de Fourier discrète.

La transformée de Fourier discrète ou **DFT** permet de décomposer une fonction quelconque, périodique ou non, en sinus et cosinus, et donc de connaître ses composantes de fréquence:

Soit le signal échantillonné  $f_k=f(kT)$  où  $T$  est la période d'échantillonnage (et  $f_s=1/T$  est la fréquence d'échantillonnage).  $N$  est le nombre de points, trouver les  $a_j$  et  $b_j$  tels que:

$$F_k = \sum_0^{N-1} a_j \cos(2\pi jk / N) + b_j \sin(2\pi jk / N)$$

Solution:  $a_j=2/N*\text{Réel}(f_j)$        $b_j=-2/N*\text{Imaginaire}(f_j)$  avec

$$f_j = \sum_{k=0}^{N-1} F(kT) e^{-2\pi jnk / N}, \text{ j}^{\text{ième}} \text{ composant de la DFT}$$

**Propriétés:**

1. -  $f_0$  = composante continue du signal

2. -  $f_j$  = composante correspondant à une fréquence de  $\frac{N}{jT} = \frac{N}{j} f_s$

3. -  $f_{N-j} = f_j$ , ( $j \neq 0$ ) et donc  $f_j$  n'est utile que si  $j \leq N/2$  qui correspond à la fréquence dite de Nyquist (moitié de la fréquence d'échantillonnage), d'où **le théorème de Shannon**, théorème fondamental de l'échantillonnage, déjà vu p 18 :

**Un signal de fréquence  $f$  échantillonné avec une fréquence d'échantillonnage  $f_s$  ne peut être reconstitué (= l'information du signal a été correctement acquise) que si :**

$$f_s > 2f$$

4. Si la fonction  $F$  est **périodique de fréquence  $s$** , alors la TFD se limite à **une série de pics** aux fréquences **multiples entiers de  $s$**  (plus éventuellement  $f_0$ ). Le pic de fréquence  $s$  est appelé *fondamental*, les pics de fréquence  $j.s$  ( $j$  entier  $>1$ ) les *harmoniques*.

Pour un signal non sinusoidal, le théorème de Shannon s'applique à toutes les composantes fréquentielles du signal. (Comme le nombre d'harmoniques est généralement infini, on ne garde que celles nécessaires pour reconstituer le signal avec une « certaine » précision.

**Par conséquent, il est normalement nécessaire, avant d'échantillonner un signal, de s'assurer que toutes les composantes de fréquence supérieure à  $f_s/2$  ont été éliminées (par filtrage dit « antialiasing »).**

Le nombre de multiplications nécessaires pour effectuer une DFT, proportionnel à  $N^2$ , devient vite colossal. Heureusement, il existe un algorithme dichotomique ramenant ce nombre à  $N \log_2(N)$ , c'est la "Fast Fourier Transform" ou FFT. Seule contrainte de la FFT,  $N$  doit être une puissance de 2. Labview utilise l'un ou l'autre, suivant le nombre de points.

On utilise en fait plus souvent le spectre en puissance du signal:

$$P_j = \frac{\left( |f_j|^2 + |f_{N-j}|^2 \right)}{N^2}, \quad P_0 = \frac{\left( |f_0|^2 \right)}{N^2}$$

qui, en vertu du théorème de Shannon, n'a de signification que jusqu'à  $j=N/2$ .

### 3. **Quelques outils disponibles :**

- a. **Outils d'acquisition avec carte National Instruments (sous-menu Acquisition de données/ Entrées analogiques) :**

*Points importants :*

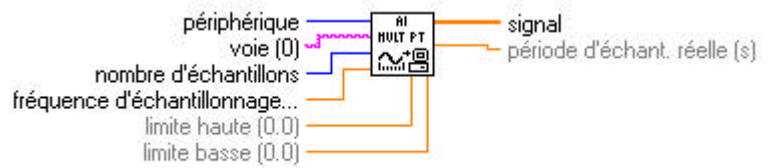
⇒ **périphérique** = numéro de la carte (il peut y en avoir plusieurs), 1 par défaut.

⇒ **Voie** = numéro de voie : de 0 à 7, c'est une chaîne de caractère contenant le numéro de voie dans le premier cas et la liste des numéros séparés par des virgules dans le second.

⇒ Nombre d'échantillons

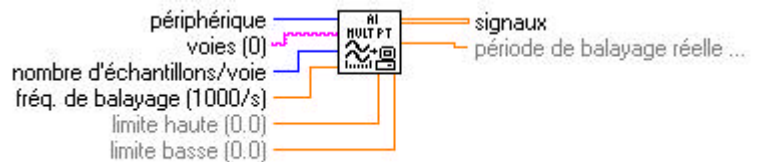
⇒ Fréquence d'échantillonnage.

Acquisition tableau de points unidimensionnel sur 1 voie :



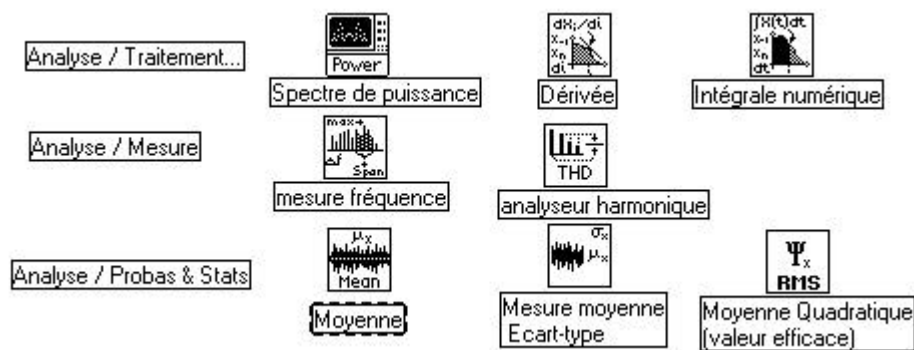
AI-Acquérir un signal.vi

Acquisition tableau de points bidimensionnel sur plusieurs voies « simultanément »:



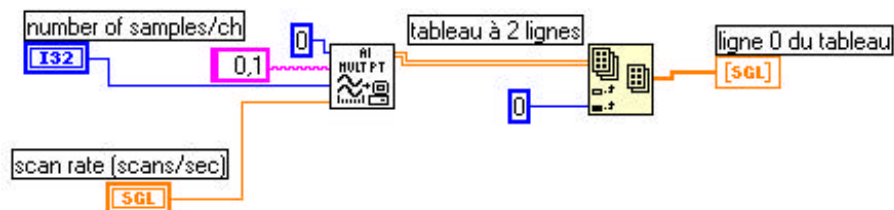
AI-Acquérir des signaux.vi

**b. Principaux outils de traitement et de mesure (liste non exhaustive, d'autres seront peut-être nécessaires...):**



*Et bien d'autres encore...*

Pour la partie détection synchrone (2eme partie) il sera nécessaire, lors de l'acquisition **simultanée** de 2 signaux, d'extraire d'un tableau bi-dimensionnel (1 ligne par voie), des sous-tableaux unidimensionnels. On utilise pour cela la fonction "Index tableau" (palette tableaux), redimensionnée grâce à un « cliquer-glisser » sur son coin inférieur gauche:



Pour pouvoir ne pas connecter une des bornes du ~~no~~nd Index tableau (et donc prendre l'intégralité de la dimension correspondante), il faut choisir "Désactiver l'indexation" dans le menu surgissant obtenu sur cette borne (ici, la première).

## 4. Vls à réaliser :

### a. VI 1: Acquisition et analyse d'un signal

L'entrée analogique 0 de la carte (Périphérique 1) est connectée à un générateur de signaux :

*Transformer le PC en oscilloscope une voie avec choix de la fréquence d'acquisition et du nombre d'échantillons. On affichera aussi le spectre de puissance gradué en Hz. On pourra afficher enfin : la fréquence fondamentale du signal, sa valeur moyenne, sa valeur efficace et tout autre information utile sur le signal.*

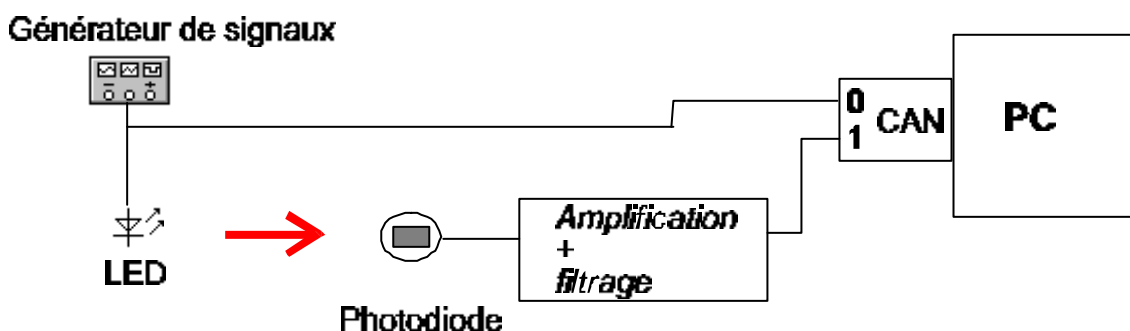
### b. VI 2: Détection synchrone:

Le signal alimentant la source lumineuse (LED : diode electro-luminescente) est celui utilisé précédemment (voie 0) .

La LED ne laisse passer le courant que dans un sens : Régler l'amplitude et l'offset du générateur afin que le signal soit compris entre 1 et 8V.

*Le signal de la photodiode est amplifié (le gain est négatif).*

Un filtrage passe haut est effectué, sa fréquence de coupure est de l'ordre de 300 Hz. Un filtrage passe bas antirepliement coupe au delà de environ 8 kHz.



*Si vous avez le temps à la fin de la première séance, essayez de réaliser la détection de la LED (indicateur booléen indiquant si la LED est visible ou non) soit à l'aide de la transformée de Fourier, soit par filtrage numérique.*

La détection synchrone, ou méthode du produit scalaire, est une méthode utilisée depuis longtemps car elle peut être traitée de façon entièrement analogique (c'est à dire alors sans calcul). Nous l'utiliserons de façon numérique:

Soient S le signal de la photodiode (voie 1) et R le signal de référence en (voie 0) : **Si tous deux sont de valeur moyenne nulle** (pas de composante continue). On a alors:

$$R = A \sin(\omega t)$$

$$S = B \sin(\omega t + \Phi) + f(t) \quad (B = \text{grandeur à mesurer})$$

f(t) est le bruit (électronique ou optique), on peut le décomposer en son spectre de fréquence (TFD):

$$f(t) = \sum_i f_i * \sin(\omega_i t)$$

$$\text{Effectuons le produit : } R * S = AB \sin \omega * \sin(\omega + \mathbf{f}) + A \sum_i f_i \sin \omega \sin \omega_i t$$

Si l'on moyenne ce signal sur un nombre suffisamment grand de points, on obtient :

$$M = \int_0^{\infty} R * S dt$$

Or on peut montrer facilement que :

$$\int_0^{\infty} \sin(\omega_1 t) \sin(\omega_2 t + \mathbf{f}) dt = \frac{1}{2} \cos \mathbf{f} \quad \text{si } \omega_1 = \omega_2$$

$$= 0 \quad \text{sinon}$$

Donc  $M \approx \frac{1}{2} AB \cos \Phi$  si f(t) a peu d'harmoniques de pulsation  $\omega$ .

Il s'agit ici d'un produit scalaire entre les 2 signaux, on obtient alors la composante X de S en phase avec R

La **valeur moyenne** de RS est donc proportionnelle à l'intensité du signal à détecter si les 2 signaux ne sont pas en quadrature et si le déphasage est constant. Nous l'appellerons « **signal de détection** ».

*En supposant dans un premier temps que le déphasage est constant et différent de  $\pi/2$ , réaliser le VI permettant de montrer l'évolution du signal de détection en fonction du temps.*

*On vérifiera l'immunité du signal aux lumières parasites, en particulier les 50 Hz redressés des néons.*

Dans un deuxième temps, on imaginera un dispositif permettant de mesurer aussi le déphasage. Indication : ce type de dispositif s'appelle alors « double détection synchrone », il consiste à effectuer, outre la détermination de la composante  $X$ , le calcul de la composante  $Y$  grâce à un « produit scalaire » avec un signal « calculé » déphasé de  $\pi/2$  avec  $R$  et de même module.

Essayer d'optimiser le système de détection, notamment la fréquence d'excitation. Le signal de phase est-il utilisable ?

A partir d'une estimation de la moyenne et de l'écart type du signal de détection mesuré lorsque la LED est invisible, on déterminera un seuil de détection au delà duquel un indicateur indiquera la visibilité de la LED avec un taux d'erreur inférieur à 0.5%.

### **Notion de seuil de détection**

Supposons que l'on veuille déterminer si un événement mesurable (mesure  $X$ ) est présent ( $X > 0$ ) ou absent ( $X = 0$ ).

On détermine alors un « seuil de décision »  $S$  (si  $X < S$ , événement absent, si  $X > S$ , événement présent)

On montre alors que :

la plus petite mesure non nulle détectable, et pour laquelle le risque d'erreur est le même pour les 2 types d'erreurs possibles (erreur A = détection alors que l'événement est absent ; erreur B = non détection alors que l'événement est présent) est **2 fois le seuil de décision**, on l'appelle **seuil de détection**.

Si  $X$  est de valeur moyenne nulle et de distribution normale, on peut montrer que le seuil de détection  $S$  est environ:

$3s$  pour un taux d'erreur de 5%

$5s$  pour un taux d'erreur de 0.5%,

où  $s$  est l'écart type de la mesure proche de zéro.



## VI. TP N°3 : LIAISONS NUMERIQUES / ETALONNAGE

### 1. Introduction :

Ce TP a 2 buts :

1 : Apprendre à dialoguer, grâce à Labview, avec un instrument numérique possédant une liaison numérique,

2 : Créer, étalonner et évaluer un instrument de mesure, ici une balance dont le capteur est une jauge de contrainte.

### 2. Préambule : Incertitude de mesure, régression linéaire

Ces notions seront vues (ou revues) plus en détail pendant les activités d'UP3.

(ou sur <http://www.emse.fr/~pbreuil/capmes/index.htm>)

Soit une variable  $x$  dont la valeur réelle (mais pas forcément connue) est  $x_R$ ,

On procède à  $n$  mesures, ou évaluations, de  $x$  appelées  $x_i$  ( $1 \leq i \leq n$ ).

Ces mesures sont généralement différentes de  $x_R$ , cette différence est la variable **erreur**  $\varepsilon$ , dont on distingue 2 composantes:  $\mathbf{e} = \mathbf{e}_A + \mathbf{e}_B$

#### a. L'erreur systématique: $\mathbf{e}_B$

La **valeur moyenne de l'erreur systématique est non nulle**: Le mesurage donne une valeur qui s'écarte systématiquement de la valeur vraie.

L'erreur systématique intervient dans la notion de **justesse** : une méthode d'analyse est juste quand on a pu éliminer l'erreur systématique. Cette élimination se fait le plus souvent à l'aide **d'étalons** qui ne doivent pas amener eux-mêmes une erreur .

D'une façon générale, on peut considérer que l'erreur systématique n'est finalement jamais évaluée car elle est:

- soit inconnue,

- soit connue et alors corrigée (par exemple par comparaison avec un étalon), auquel cas on l'annule.

### b. L'erreur accidentelle ou aléatoire ( $e_A$ )

C'est une variable **aléatoire**. Lorsqu'on répète plusieurs fois le mesurage d'une grandeur physique ou chimique constante, on obtient généralement différentes valeurs plus ou moins dispersées (et qui sont souvent distribuées suivant une loi normale, voir plus loin).

Par définition, la **valeur moyenne** de l'erreur accidentelle, ou aléatoire, **est nulle**.

Dans la plupart des cas, les erreurs accidentelles ont une **distribution normale**. Cette hypothèse de distribution normale, valable pour 99% des cas, provient du fait que plusieurs sources indépendantes contribuent généralement à cette erreur. Or le *Théorème Central Limite* nous dit qu'une combinaison linéaire d'un nombre suffisamment grand de variables indépendantes de distributions quelconques tend vers une distribution normale.

### c. Caractérisation de la mesure et de son erreur:

On peut caractériser la mesure par:

- **une moyenne estimée** de la mesure qui est la moyenne arithmétique des  $n$  mesurages  $x_1, x_2, \dots, x_i, \dots, x_n$  faits pour caractériser une grandeur :

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Si l'**erreur systématique est nulle** (ce que l'on suppose pour la suite), alors  $\bar{x} \rightarrow x_R$  quand  $n \rightarrow \infty$  ( $x_R$  = valeur **réelle**, à priori **inconnue**).

- **un écart-type** défini comme étant la racine carrée de la moyenne du carré de l'écart entre la mesure et la valeur réelle  $x_R$ :

$$s = \sqrt{\overline{(x_i - x_R)^2}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x_R)^2} \quad (\text{fonction Excel ECARTYPEP})$$

Mais généralement,  $x_R$  est inconnu, on en a juste une estimation par la moyenne  $\bar{x}$ . On peut alors calculer une estimation de l'écart type notée  $s$ :

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (\text{fonction Excel ECARTYPE})$$

De même, l'écart type estimé tend vers l'écart type réel quand  $n \rightarrow \infty$ .

- **la variance** est égale au carré de l'écart-type :

$$V = s^2$$

Si la distribution des erreurs est **normale** (ou **gaussienne**) (c'est généralement le cas), on peut alors calculer, à l'aide du tableau des **coefficients de Student**, la probabilité  $p$  pour qu'une mesure individuelle soit hors de l'intervalle  $[\bar{x} - ts, \bar{x} + ts]$

**Coefs de Student t**

intervalle de conf.	90.0%	95.0%	98.0%	99.0%	99.9%
p	0.1	0.05	0.02	0.01	0.001
deg lib					
1	6.31	12.71	31.82	63.66	636.58
2	2.92	4.30	6.96	9.92	31.60
3	2.35	3.18	4.54	5.84	12.92
4	2.13	2.78	3.75	4.60	8.61
5	2.02	2.57	3.36	4.03	6.87
6	1.94	2.45	3.14	3.71	5.96
7	1.89	2.36	3.00	3.50	5.41
8	1.86	2.31	2.90	3.36	5.04
9	1.83	2.26	2.82	3.25	4.78
10	1.81	2.23	2.76	3.17	4.59
12	1.78	2.18	2.68	3.05	4.32
14	1.76	2.14	2.62	2.98	4.14
17	1.74	2.11	2.57	2.90	3.97
20	1.72	2.09	2.53	2.85	3.85
30	1.70	2.04	2.46	2.75	3.65
40	1.68	2.02	2.42	2.70	3.55
50	1.68	2.01	2.40	2.68	3.50
100	1.66	1.98	2.36	2.63	3.39
100000	1.64	1.96	2.33	2.58	3.29

t.s est appelé incertitude élargie et t est le coefficient de Student .

La dernière ligne (n grand) est calculée à partir de l'intégrale de la Gaussienne (fonction erf)

Dans le cas de l'estimation de l'écart-type, le nombre de degrés de liberté est n-1.

Ainsi, pour n suffisamment grand (n>20), 95% des mesures sont dans l'intervalle  $[\bar{x} - 2s, \bar{x} + 2s]$ , et 99.7% dans l'intervalle  $[\bar{x} - 3s, \bar{x} + 3s]$

**Rappelons que si l'on prend maintenant pour mesure la moyenne de n mesures individuelles dont les erreurs sont aléatoires et indépendantes, alors la valeur de l'écart type**

**devient :  $s_{\bar{x}} = \frac{s_x}{\sqrt{n}}$  (cf théorème de la moyenne, p 19)**

Le résultat d'une mesure doit comporter **4 éléments** :

$$\text{Ex : } C_{\text{NO}} = \underset{1}{125.3} \underset{2}{\text{ppb}} \pm \underset{3}{1.7} \underset{4}{\text{ppb}} \quad (k=2)$$

**1 : Valeur numérique** avec un nombre correct de décimales

## 2 : Unité

3 : **Incertitude élargie** =  $t \cdot \sigma$  (= intervalle de confiance x 2)

4 : **Le coefficient d'élargissement t utilisé** (généralement noté k : ex k=2)

Si la mesure est non nulle, on utilise souvent **l'incertitude (élargie) relative (à la mesure)**, calculée à partir de l'écart type relatif :  $s_r = \frac{s}{|x|}$ , souvent exprimée en pourcentage de la mesure.

### d. Evaluation de l'incertitude :

On distingue 2 types de méthode :

**Evaluation de type A :** par analyse statistique de séries de mesures, à l'aide des formules du paragraphe précédent

**Evaluation de type B :** « Par tout autre moyen » (!) : généralement, on évalue l'effet sur l'incertitude finale des différentes sources d'incertitude, elles même évaluées :

- Par une méthode de type A,
- Par des données constructeur, d'étalonnage etc...

*Remarque: L'incertitude donnée par le constructeur peut être **constante pour un appareil donné**. Mais on ne connaît ni sa valeur, ni même son signe, elle sera donc considérée comme une erreur aléatoire.*

On utilise alors **les lois de propagation des écart-type** (ou des incertitudes, au facteur d'élargissement près) :

Si la mesure finale, y, est fonction de variables  $x_i$ , **indépendantes** dont les  $\sigma(x_i)$  sont connus:

$y=f(x_1 \dots x_n)$ , alors l'écart type de y s'écrit :

$$s^2(y) = \sum_1^n \left( \frac{\partial f}{\partial x_i} \right)^2 s^2(x_i)$$

(Il s'agit en fait du théorème des accroissements finis appliqué à la variance)

**Exemples :** (variables indépendantes, donc pas de terme de covariance)

1 : **Combinaison linéaire**  $y = \sum_i a_i x_i$ , alors  $s^2(y) = \sum_i (a_i s(x_i))^2$

Ex: somme ou différence:  $y = a + b - c$ , alors  $s(y) = \sqrt{s^2(a) + s^2(b) + s^2(c)}$

2 : **Produits et puissances** :  $y = A \prod_i x_i^{a_i}$

Il est dans ce cas plus commode de considérer l'écart type relatif  $\frac{s(y)}{|y|}$  :

$$\text{On a alors : } \left( \frac{s(y)}{y} \right)^2 = \sum_i \left( a_i \frac{s(x_i)}{x_i} \right)^2$$

$$\text{Ex : } y = K \cdot \frac{a^2 b}{c} \text{ où } K = \text{constante, alors } \frac{s(y)}{|y|} = \sqrt{\left( \frac{2s(a)}{a} \right)^2 + \left( \frac{s(b)}{b} \right)^2 + \left( \frac{s(c)}{c} \right)^2}$$

$$\text{Ou encore: } s(y) = K \sqrt{\left( \frac{2ab}{c} s(a) \right)^2 + \left( \frac{a}{c} s(b) \right)^2 + \left( \frac{a^2 b}{c^2} s(c) \right)^2}$$

Rappelons, pour mémoire (cf p 18), que l'écart-type de l'erreur due à la discrétisation s'écrit :  $s_d = \frac{d}{2\sqrt{3}} = 0.29d$ , où  $d$  est la résolution.

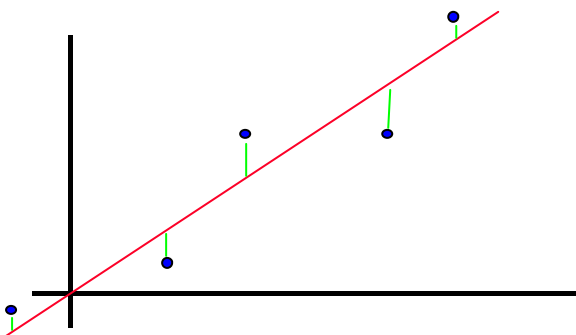
### e. L'étalonnage par régression linéaire

L'étalonnage (*calibration* en anglais) d'un instrument de mesure ou d'un capteur, consiste à **modéliser le signal de sortie du capteur** (appelé Xvariable, généralement un signal électrique) en fonction de la variable mesurée (appelée Yvariable, variable physico-chimique). Pendant l'étalonnage, les Yvariables, considérées comme étalons, sont connues ou évaluées, ainsi que les Xvariables associées.

Le modèle est alors une fonction F:  $\hat{Y} = F(X)$

L'utilisation normal du modèle, ou prédiction, permet d'évaluer les  $\hat{Y}$  à l'aide des seuls X et du modèle F.

Si l'on fait **l'hypothèse d'une loi linéaire** reliant les X et les Yvariables, alors le modèle peut être déterminé par **une régression linéaire** :



Cette méthode permet de déterminer l'équation de la droite (pente  $a$  + ordonnée à l'origine  $b$ ) minimisant la somme des carrés des écarts entre mesure et valeurs calculées  $\hat{y} = ax + b$  :

$$S = \sum (y - \hat{y})^2 \text{ minimale}$$

**En appliquant cette méthode, on sup-**

pose implicitement que seules les Y variables ont des erreurs aléatoires.

Les calculs de régression linéaire seront faits à l'aide d'Excel qui offre 2 possibilités :

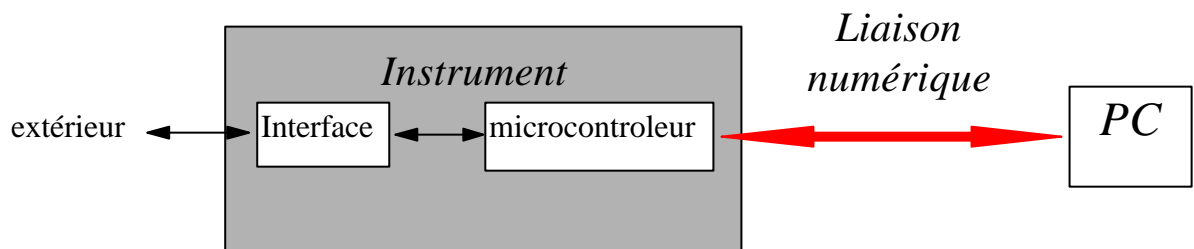
- **courbe de tendance**, à partir d'un graphe « nuage de points »
- **fonction DROITEREG**, plus complexe, mais permettant de calculer les écart-type de la pente et de l'ordonnée à l'origine.

On se reportera à l'aide en ligne Excel pour plus de précisions.

### 3. Utilisation d'une liaison série RS232 :

Les liaisons numériques permettent à certains instruments numériques (multimètre de table, data logger, régulateur etc...) de dialoguer avec un Micro-ordinateur ou avec d'autres instruments.

#### a. Généralités :



Il existe, en instrumentation, de nombreuses formes physiques de liaisons numériques (RS232, RS485, GPIB, ethernet, CAN bus etc...) et pour chaque forme de nombreux protocoles, souvent propres à chaque constructeur.

Nous allons étudier la forme de dialogue la plus simple à mettre en œuvre qui le dialogue « Maître – esclave ». Dans cette configuration, l'instrument, l'esclave, attend des ordres émis par le maître (ici un PC) pour y répondre.

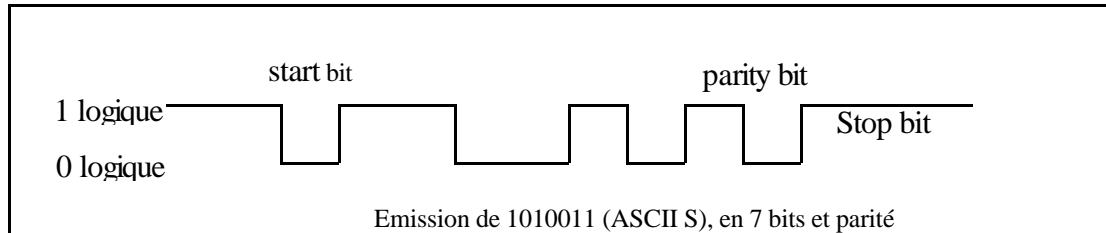
Ces dialogues se font souvent en ASCII, c'est à dire à l'aide de chaînes de caractères. Du point de vue informatique, une chaîne de n caractères est un tableau de n octets (entier entre 0 et 255), la valeur de chaque octet correspondant au code « ASCII » d'un caractère.

Les codes des caractères dits imprimables (lettres, chiffres et ponctuation) sont compris entre 32 et 127. Les codes 0 à 31 sont des caractères de contrôle (ex : 13=CR=Carriage Return, 27=Esc=Escape...).

La liaison numérique utilisée pour ce TP est une liaison série RS232 qui a l'avantage d'être présente sur tous les PC.

C'est une liaison série (1 seul fil pour les données), asynchrone (pas de synchronisation avec un signal d'horloge), full duplex (un fil par sens). Elle est peu performante en vitesse mais facile à mettre en œuvre et présente sur tous les PC.

Transmission d'un octet:



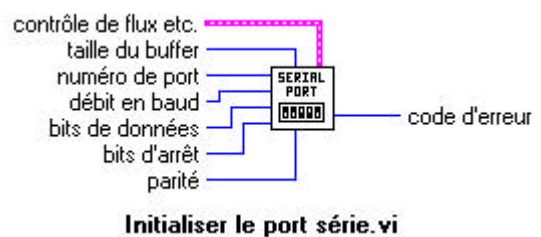
**b. Outils Labview utilisables : dans Communication \E/S d'instruments /Série**

Points communs :

- Numéro de port : numéro du port série, 0 pour COM1 et 1 pour COM2
- Code d'erreur : 0 si pas d'erreur d'exécution ou de communication, voir aide en ligne sinon.

**Initialisation :** les 2 entités communicantes doivent avoir des réglages identiques :

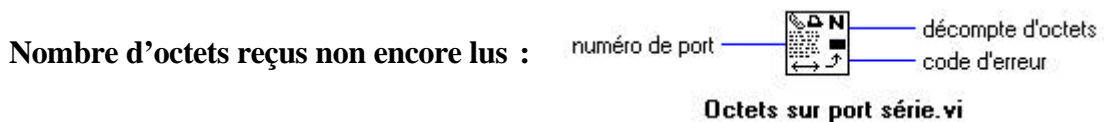
- vitesse de 60 à 19200 bauds (bits/s)
- bits de données = 7 ou 8
- parité : aucune, paire ou impaire
- nombre bits d'arrêt : généralement 1



*Buffer* : zone de mémoire du PC permettant d'accueillir temporairement les octets émis par l'instrument même si le programme recevant n'est pas prêt (sinon risque de perte de données), typiquement quelques centaines d'octets.

**Ecriture chaîne :** L'utilisateur a la charge d'ajouter éventuellement le ou les caractères de terminaison (généralement CR, LF ou CR+LF)





**Lecture chaîne :** assez limité ! : lecture d'une chaîne faisant au maximum n (=nombre d'octets demandé) caractères, si l'instrument envoie une chaîne de moins de n caractères, Labview attend...



L'utilisateur doit donc détecter le (ou les) caractères de fin de chaîne, le moyen le plus sûr étant de lire les caractères 1 par 1 et de les concaténer jusqu'à détection du caractère de fin de chaîne (à l'aide d'une boucle « while » munie d'un registre à décalage).


### c. Protocole :

L'instrument étudié est un « module d'acquisition » de type 6B11 d'Analog Device. Ces modules permettent de faire l'acquisition d'une tension avec une bonne précision. Plusieurs gammes de mesure sont disponibles et programmables (de +/- 15 mV à +/- 5 V).

**La durée d'une conversion analogique digitale est 60 ms.**

**Le module est configuré pour une vitesse d'acquisition de 19200 bauds** (~ bits / s) (8 bits de données, 1 bit stop, pas de parité, pas de protocole), une adresse 00 et pas de Checksum. La gamme de mesure est réglée sur +/- 15 mV . Ne pas tenter de modifier ces paramètres.

Protocole : extrait de la notice Analog Device relative à la lecture d'une mesure

- Commande : #AA↵ avec :
  - # → caractère « # » = code de la commande « Emission de la mesure la plus récente »,
  - AA → adresse du module en hexadécimal sur 2 caractères (00 si adresse 0),
  - ↵ → caractère de contrôle « carriage return » (code ASCII 13) prédéfini dans Labview :  )
- Réponse : >(data)↵ avec :
  - > → caractère de début de message,
  - (data) → chaîne de caractères représentant le réponse sous forme décimale,
  - ↵ → carriage return (fin de message)

- Exemple : Commande #00 ↵ → réponse : >+4.7653 ↵ (4.7653 mV)

## 4. Première réalisation: multimètre

1 : Réaliser un instrument (multimètre numérique) permettant de lire et d'afficher graphiquement la tension présente à l'entrée.

On gèrera les erreurs de communication les plus courantes qui sont l'erreur d'initialisation et le time out (absence de réponse au bout d'un certain temps).

Quel est, en mV, la résolution du convertisseur analogique-numérique ?

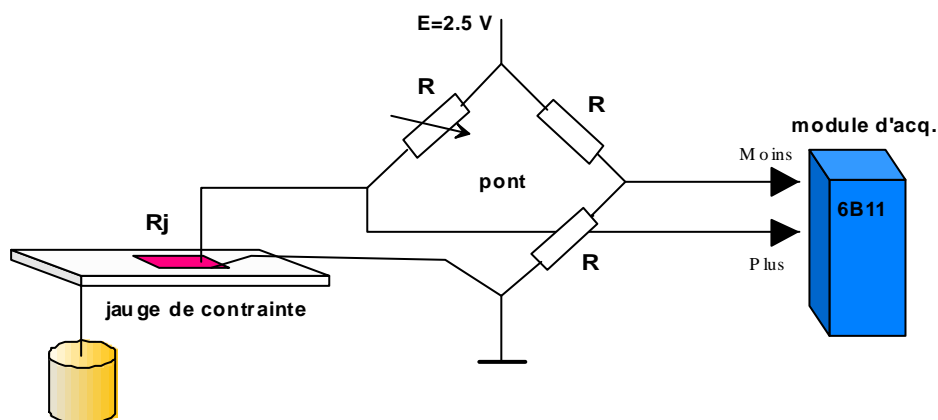
Etudier l'influence du nombre de mesures moyennées sur l'écart type de la mesure (cf théorème de la moyenne, p 19).

Vérifier notamment le type de la loi  $S(\bar{x}) = f(n)$ , correspond-il à la théorie ?  
Quelle peut être l'influence d'une éventuelle temporisation entre 2 mesures ?

On choisira pour la suite un nombre de mesures correspondant à un temps de mesure total de l'ordre de 2 à 3 s.

## 5. Deuxième réalisation : balance numérique

L'entrée du module est en fait reliée à un système de mesure de masse :

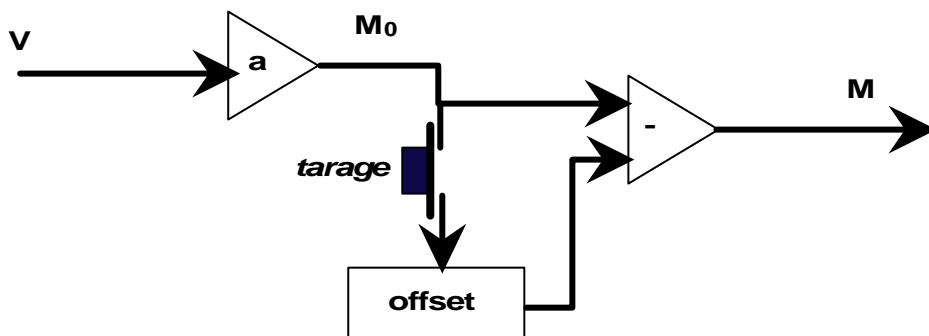


**On suppose ici que vous avez déterminé une méthode de mesure (moyenne sur n mesure) qui deviendra pour la suite LA mesure.**

Afin de réaliser une balance numérique indiquant la masse mesurée en kg, procéder à un étalonnage à l'aide d'un certain nombre de mesures (20 à 40) à l'aide de masses calibrées utilisées dans un ordre aléatoire (pourquoi ?), on utilisera un minimum de 5 ou 6 valeurs de masse différentes. Ce sont les 20 à 40 mesures qui seront utilisées individuellement pour la régression linéaire.

Pour faire les calculs statistiques (écart types, régressions linéaires ou non linéaires...), on pourra utiliser Excel.

A l'aide du modèle, réaliser un affichage en masse, ajouter enfin un bouton de tarage permettant à tout moment de ramener la mesure à 0 (l'estimation de l'ordonnée à l'origine b devient alors inutile).



$M_0 = aV$ , donc le calcul de l'écart type se fait de la façon suivante :

$$S_{M_0} = \sqrt{a^2 S_V^2 + V^2 S_a^2} \quad (\text{car } S(f(x_1, \dots, x_n))^2 = \sum_1^n \left( \frac{\partial f}{\partial x_i} \right)^2 S(x_i)^2)$$

avec :

$V$  = mesure de la tension, l'incertitude dépendra donc du point de fonctionnement,

$a$  = pente déterminée par régression linéaire,

$\sigma_V$  = écart type estimé de la mesure de tension, déterminé à l'aide de plusieurs mesurages d'une même masse dans les conditions normales d'utilisation, on vérifiera si cet écart type est constant sur toute la gamme,

$\sigma_a$  = écart type estimé de la pente, donné par la fonction Excel « DROITEREG » (voir aide Excel).

De plus,  $M = M_0 - \text{offset}$ , offset étant une valeur de  $M_0$  mémorisée lors d'un tarage, donc :

$$S_M = \sqrt{S_{M_0}^2 + S_{\text{offset}}^2} \approx \sqrt{2} S_{M_0} \text{ si } S_{M_0} \text{ constant sur toute la gamme.}$$

*Indiquer l'incertitude « à 95% » sur la mesure de la masse, avec les conditions de fonctionnement.*