

Chapter 3

Un exemple de traitement.

Pour cet exemple nous allons traiter un modèle relevant de la "Dynamique des Systèmes" au sens de Jay W. Forrester. Il s'agit du modèle "Future Electronics" dont l'énoncé figure en annexe. Pour pouvoir transcrire ce modèle sous Matlab il est nécessaire de donner des éléments de *Dynamique des Systèmes*.

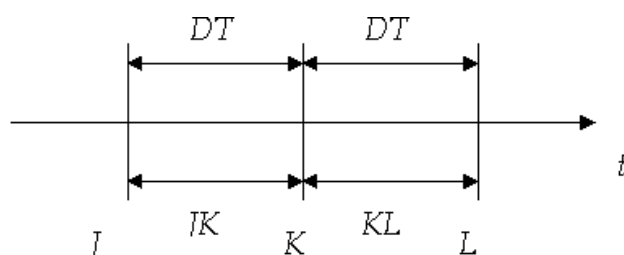
3.1 Eléments de Dynamique des Systèmes.

3.1.1 Echelle de temps et "indices".

Modèle de SSD = modèle de résolution numérique d'équations
différentielles
discrétisées à l'aide d'un processus de calcul pas.



On ne connaît que le "temps discret".



Echelle de temps discret en SSD (t_{SSD2})

A tout moment on ne connaît que 3 instants :

instant présent $\rightarrow$ K
instant immédiatement antérieur $\rightarrow$ J
instant immédiatement postérieur $\rightarrow$ L

Instants régulièrement espacés, valeur numérique de l'espacement = **DT**

↓

Paramètre propre au processus de calcul
= pas de calcul, intervalle de solution, pas d'intégration

↓

Sa valeur obéit à une règle empirique très importante.

3.1.2 La symbolique d'écriture des équations. Symbolique scripturale.

3.1.2.1 Les variables.

A l'origine $\longrightarrow$ symboles alphanumériques d'au plus 6 caractères.
symbole suivi d'un point et d'un "post indice de temps" (sauf pour les constantes)

Exemples

NIVEAU.K, N1.K, FLUX.KL, C2, CTE

3.1.2.2 Les équations.

Equation de niveau.

Toutes les équations de niveau ont le même format d'écriture.

$$N.K = N.L + DT \times (FE.JK - FS.JK) \quad (1,L)$$

$$N = 100 \quad (1.1,N)$$

N : Niveau (unités)

FE : Flux d'Entrée (unités / unité de temps)

FS : Flux de Sortie (unités / unités de temps)

Commentaires.

$\longrightarrow$ Equation de niveau proprement dite est suivie par son numéro (ici 1 puisque c'est la première équation présentée), une virgule et une "lettre type" qui permet de qualifier l'équation (L = Level).

→ “ **Le pas de calcul DT apparaît dans toutes les équations de niveau et dans aucune autre.** ” (Principe 5.3-1).

→ Equation suivie nécessairement d’une **équation d’initialisation**. [nécessaire pour la résolution numérique].

Numéro de l’équation d’initialisation donné sous forme d’une subdivision décimale du numéro de son équation de niveau (ici 1.1).

Suivi d’une virgule et de la lettre type N, pour iNitial.

→ On donne toujours après l’équation, la signification des symboles employés

$\Downarrow$
 Nom complet des variables
 + (unités dans lesquelles sont mesurées ces variables)

Quand on élabore un modèle de SSD on doit se livrer à chaque fois à une “ **analyse dimensionnelle** ” $\Leftrightarrow$ on ne peut “ additionner des choux et des carottes “.

$$\begin{aligned}
 \text{Unités} &= \text{Unités} + (\text{Unité de temps}) \cdot \frac{\text{Unités}}{\text{Unité de temps}} \\
 \text{Niveau} &= \text{Niveau} + DT \times \text{Flux}
 \end{aligned}$$

→ Il n’y a pas de limite au nombre de flux qui peuvent agir sur un niveau.

Equation de flux.

Equation de politique disant comment un niveau se modifie.

En général c’est une fonction de niveaux et de constantes.

$$\text{Flux} = f(\text{Niveaux}, \text{constantes})$$

ou

$$F = \left(\sum_{i=1}^p N_i \right) \times \left(\frac{1}{DEL} \right) \times A_1 \times M + A_2$$

$N \longrightarrow$ variable d’état
$F \longrightarrow$ flux
$A \longrightarrow$ auxiliaire
$DEL \longrightarrow$ délai de temps
$M \longrightarrow$ “modulateur”, (fonction d’interpolation linéaire)

Exemple.

$$\begin{aligned}
 F.KL &= NIV.K \times NORMTC \times MULT.K & (2,R) \\
 NORMTC &= 0.10 & (2.1,C)
 \end{aligned}$$

F : Flux (unités par unité de temps)

$NORMTC$: Norme Taux de Croissance (fraction par unité de temps)

$MULT$: Multiplicateur (sans dimension)

Commentaires.

→ N° 2, suivi de la lettre type R pour Rate.

→ Il y a une constante $NORMTC \Rightarrow$ valeur numérique donnée dans équation de N° = subdivision décimale du numéro de l'équation dans laquelle la constante apparaît pour la première fois. Ce numéro est suivi de la lettre type C pour Constant.

(Il y a autant d'équations que de constantes !)

→ L'équation de flux comme toutes les autres équations est suivie de l'explicitation des variables employées avec mention de l'unité de mesure

→ En partie gauche de l'équation on a la variable $MULT.K$ qui représente un type de variable non encore présenté à savoir une **variable auxiliaire**, terme apparaissant d'ailleurs dans la forme générale d'une équation de flux.

Equation auxiliaire.

Equation auxiliaire = subdivision algébrique d'une équation de flux ayant un sens précis et un rôle propre.

Exemple.

Modèle Dynamique urbaine de Forrester → variable d'état "Nouvelles Entreprises", NE

Le flux de construction de nouvelles entreprises est défini par

$$CNE.KL = CDNE.K \times RTC.K \quad 101, R$$

CNE : Construction de Nouvelles Entreprises (unités productives/an)

$CDNE$: Construction Désirée de Nouvelles Entreprises (unités productives/an)

RTC : Rapport Travailleurs dans la Construction (sans dimension)

$CDNE$ et $RTC \rightarrow$ variables auxiliaires

$$\begin{aligned}
CDNE.K &= (NCNE) \times (FCNE \times NE.K + FCAM \times AM.K + FCID \times ID.K) \times ME.K \\
&\quad + PCNE.K \quad 102, A \\
FCNE &= 1 \quad 102.1, C \\
FCAM &= 0.5 \quad 102.2, C \\
FCID &= 0.3 \quad 102.3, C \\
NCNE &= 0.05 \quad 102.4, C
\end{aligned}$$

CDNE : Construction Désirée de Nouvelles Entreprises (unités productives/an)
NCNE : Norme Construction de Nouvelles Entreprises (fraction/an)
FCNE : Facteur construction de nouvelles entreprises (sans dimension)
NE : Nouvelles Entreprises (unités productives)
FCAM : Facteur Construction des Affaires Mûres (sans dimension)
AM : Affaires Mûres (unités productives)
FCID : Facteur Construction des Industries en Déclin (sans dimension)
ID : Industries en Déclin (unités productives)
ME : Modulateur d'entreprises (sans dimension)
PCNE : Programme Construction de Nouvelles Entreprises (Unités productives/an)

ME et $PCNE \longrightarrow$ auxiliaires

Les équations auxiliaires permettent de remonter une chaîne d'information, jusqu'aux composante premières de l'information qui sont les niveaux et les constantes.

Des auxiliaires particulières les modulateurs

Un modulateur correspond à une fonction, linéaire, ou non, discrétisée. Cette fonction sert à moduler un processus "normal" en fonction de l'état du système global décrit en général par le biais d'une variable auxiliaire.

On les retrouve dans *Simulink* sous la forme de

"Look Up Table"

Exemple.

$$CNE = CDNE \times RTC$$

$CDNE \rightarrow$ processus normal de construction

$RTC \rightarrow$ modulateur exprimant l'influence des travailleurs disponibles

$$RTC = f(RTE)$$

RTE $\longrightarrow$ Rapport Travailleurs / Emplois

Modulateur $\Rightarrow$ appel à une fonction d'interpolation linéaire.

Dans Dynamo il y a deux fonctions d'interpolation $\begin{matrix} TABLE \\ TABHL \end{matrix}$

Codage de la fonction TABLE en DYNAMO

$$\begin{aligned} M &= \text{TABLE}[MT, A, \text{Min}(A), \text{Max}(A), \text{Incr}(A)] \quad N^\circ, A \\ MT &= m_1/m_2/\dots/m_n \quad N^\circ.1, T \end{aligned}$$

où

M = modulateur (extrant fonctionnel).

MT = modulateur tabulé (nom de la "table" sur laquelle la fonction opère).

A = variable intrant (variable indépendante : niveau ou auxiliaire).

$\text{Min}(A)$ = valeur minimale de A à laquelle correspond le 1er élément de la table.

$\text{Max}(A)$: valeur maximale de A à laquelle correspond le dernier élément de la table.

$\text{Incr}(A)$ = pas servant à se "déplacer" de $\text{Min}(A)$ à $\text{Max}(A)$.

$\longrightarrow$ L'axe des abscisses a des caractéristiques fixées par 3 paramètres

$$\begin{aligned} &\text{Min}(A) \\ &\text{Max}(A) \\ &\text{Incr}(A) \end{aligned}$$

$\longrightarrow$ Aux valeurs discrètes en abscisse correspondent autant de valeurs discrètes en ordonnée, celle-ci sont données dans la "table", le "vecteur" MT .

$$\begin{aligned} \text{Min}(A) &\rightarrow m_1 \\ \text{Min}(A) + \text{Incr}(A) &\rightarrow m_2 \\ \text{Min}(A) + 2 \times \text{Incr}(A) &\rightarrow m_3 \\ &\vdots \\ \text{Max}(A) = \text{Min}(A) + (n-1) \times \text{Incr}(A) &\rightarrow m_n \end{aligned}$$

Nombre d'éléments de MT donné par

$$n = \frac{\text{Max}(A) - \text{Min}(A)}{\text{Incr}(A)} + 1$$

Exemple.

$$\text{Min}(A) = 0, \quad \text{Max}(A) = 2, \quad \text{Incr}(A) = 0.5 \Rightarrow n = \frac{2-0}{0.5} + 1 = 5$$

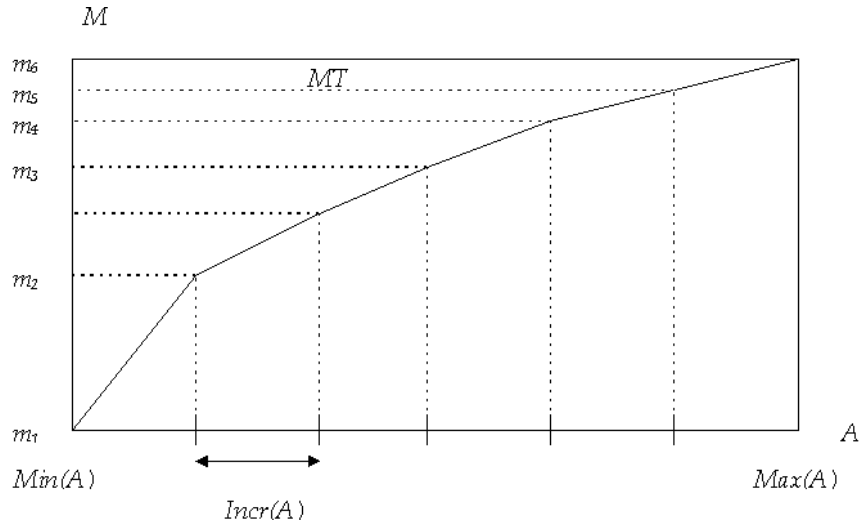


Schéma de principe d'un modulateur. (TSSD3)

De façon générale si

$$i \in [1, n]$$

$$A_i \in [Min(A), Max(A)]_{Incr(A)}$$

on a la formule d'interpolation linéaire

$$M = m_i - (A - A_i) \frac{m_{i+1} - m_i}{A_{i+1} - A_i}$$

Distinction TABLE/TABHL dans DYNAMO.

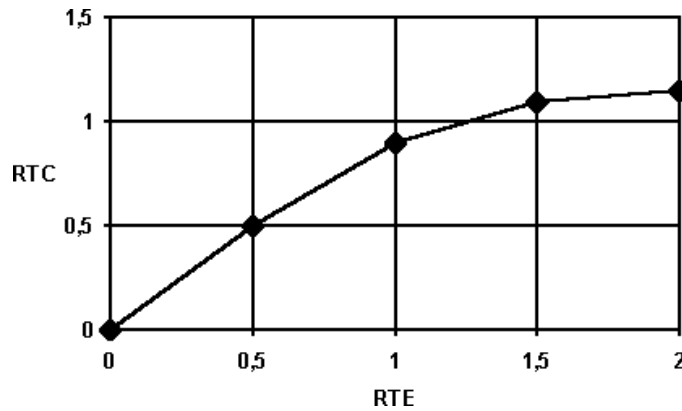
TABLE	TABHL
$\Rightarrow$ formulation restrictive	laisse plus de latitude
Si $A \notin [Min(A), Max(A)]$	si $A < Min(A) \Rightarrow M = m_1$
	$A > Max(A) \Rightarrow M = m_n$
processus de simulation bloqué	$\equiv$ "TABulation Hors Limites"

Exemple de RTC dans Dynamique Urbaine.

$$RTC.K = TABHL(RTCT, RTE.K, 0, 2, 05) \quad 138, A$$

$$RTCT = 0/0.5/0.9/1.1/1.5 \quad 138.1, T$$

En général chaque fois qu'un modulateur est défini on lui adjoint sa représentation graphique. Dans le cas présent on a



Le modulateur RTC dans Dynamique urbaine (T_{SSD4})

Les équation “supplémentaires.

Equations permettant d’obtenir une variable particulière qui nous intéresse, sans que pour autant la valeur de cette variable intervienne dans le calcul d’autres équations.

Elle est suivie de la lettre type S en Dynamo.

3.1.3 Les diagrammes utilisés en SSD

2 types Diagrammes de structure causale
 Diagrammes de flux

3.1.3.1 Les diagrammes de structure causale

Ce sont des diagrammes fléchés.

Les flèches servent à traduire une relation de cause à effet entre deux variables

La signature (+/-) traduit le type de causalité.

Fondamentalement ce type de diagramme sert à identifier et à organiser les principales boucles de rétroaction du système étudié avec leurs composantes, sans préciser la nature des variables interconnectées.

Exemple emploi migration. On considère un système urbain simplifié dans lequel on a les variables suivantes

Emplois Disponibles (ED) : nombre total d'emplois vacants

Migration (M) : migration nette de travailleurs dans l'aire urbaine

Travailleurs (T) : nombre total de travailleurs résidant dans l'aire

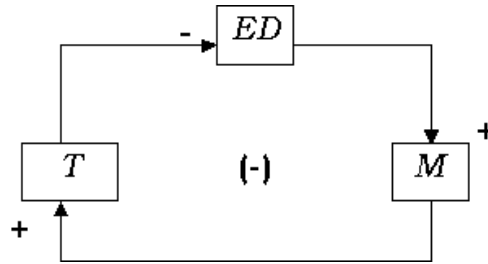
On a alors les relations binaires fléchées

$$ED \xrightarrow{+} M \quad (ED \uparrow \Rightarrow M \uparrow)$$

$$M \xrightarrow{+} T \quad (M \uparrow \Rightarrow T \uparrow)$$

$$T \xrightarrow{-} ED \quad (T \uparrow \Rightarrow ED \downarrow)$$

D'où la boucle de rétroaction (polarité globale $\Rightarrow$ appliquer la règle des signes)

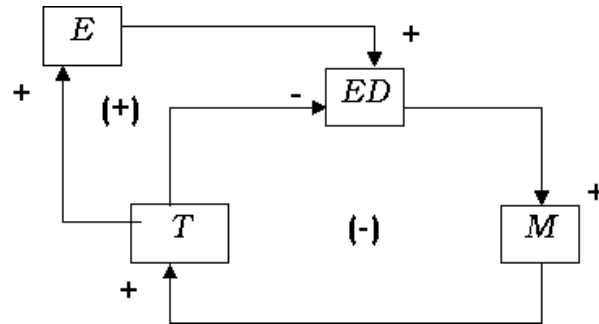


(TSSD9)

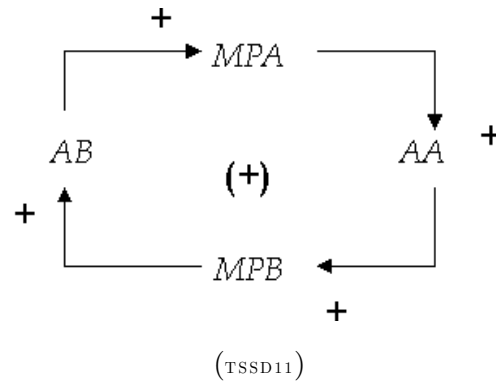
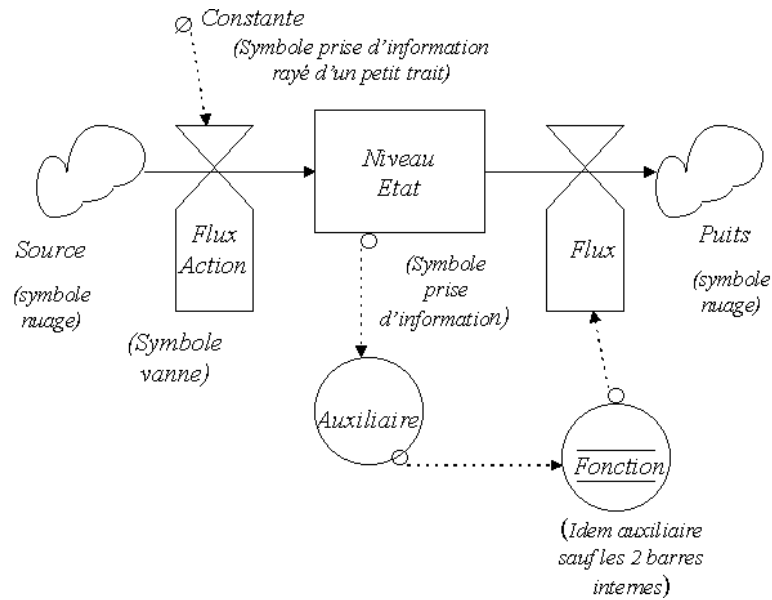
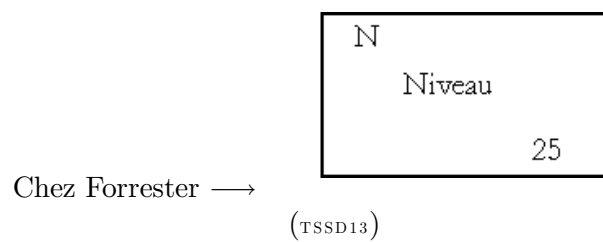
On peut introduire une nouvelle variable E , nombre total d'emplois vacants telle que

$$E \xrightarrow{+} ED \quad T \xrightarrow{+} E$$

on obtient un nouveau diagramme causal comprenant deux boucles.



(TSSD10)

Exemple de la course aux armements.**3.1.3.2 Les diagrammes de flux.****Symbolique graphique d'un diagramme de flux. (TSSD12)**

Passage d'un diagramme de structure causale à un diagramme de flux.

Exemple de la course aux armements.

→ Passer aux variables de niveau et de flux

→ Ajouter ou transformer des variables.

2 variables de niveau évidentes → Armement de A (AA)
Armement de B (AB)

Il faut **traduire** les variables “Menace perçue par...”.

[H1] Chaque niveau se remplit par un flux sans se vider.

↓

2 flux → Flux d'Armement de A (FAA)
Flux d'Armement de B (FAB)

[H2] Chaque flux est directement proportionnel au niveau d'armement de l'adversaire.

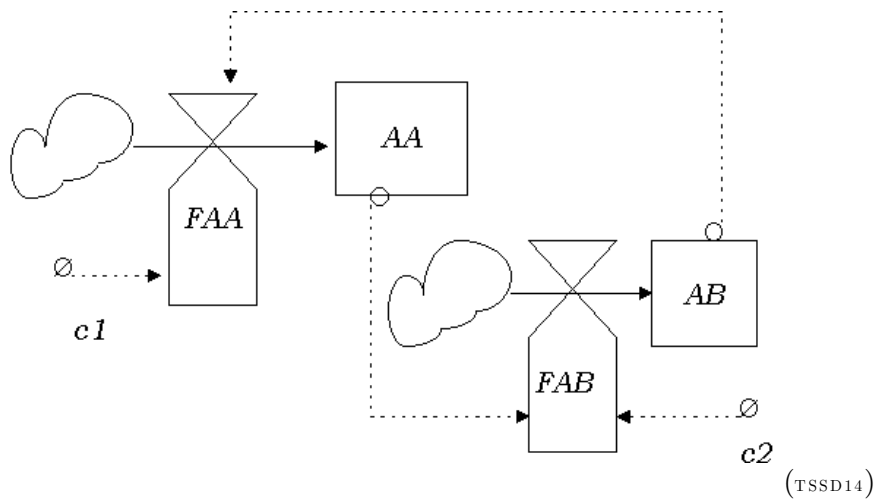
($\equiv$ traduction de la menace)

[H3] Initialement $AA = 100$, $AB = 50$

↓

$AA = AA + DT * FAA \quad 1, L \quad AB = AB + DT * FAB \quad 3, L$
 $AA = 100 \quad 1.1, N \quad AB = 50 \quad 3.1, N$
 $FAA = c1 * AB \quad 2, R \quad FAB = c2 * AA \quad 4, R$
 $c1 = 0.01 \quad 2.1, C \quad C2 = 0.02 \quad 4.1, C$

Variable	Unités
AA et AB	<i>Armement</i>
FAA et FAB	$\frac{\text{Armement}}{\text{an}}$
$c1$ et $c2$	$\frac{\text{Fraction}}{\text{an}}$



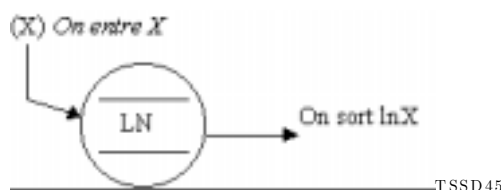
3.1.4 Fonctions et assemblages.

Les fonctions que nous allons traiter maintenant sont héritées des fonctions du compilateur DYNAMO. Elles servent à représenter des opérations particulières. On distingue deux grands types de fonctions, les fonctions sans intégration et les fonctions avec intégration. Ces dernières sous entendent la création de niveau pour les besoins d'une représentation particulière. On peut les assimiler à des macro assemblages.

3.1.4.1 Les fonctions sans intégration.

Toutes ces fonctions sont symbolisées, dans les diagrammes de flux usuels, par un cercle contenant deux barres horizontales avec entre elles le nom de la fonction.

3.1.4.1.1 Fonctions calcul. Ce sont les fonctions mathématiques usuelles telles que : l'exponentielle $\text{EXP}(X)$, le logarithme népérien $\text{LN}(X)$, la racine carrée $\text{SQRT}(X)$. D'où, par exemple le symbole



3.1.4.1.2 Les fonctions d'interpolation. Les modulateurs. Nous avons déjà présenté ces fonctions antérieurement (Cf. supra). Aussi nous n'y reviendrons pas.

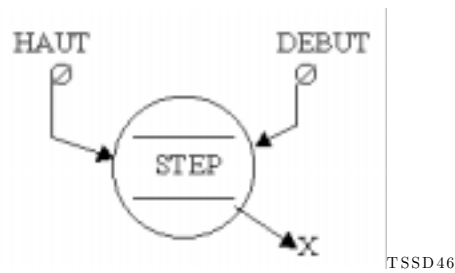
3.1.4.1.3 Les fonctions tests. D'une façon générale ces fonctions servent à tester un modèle en fournissant une excitation exogène à laquelle le système modélisé réagit, ce qui apporte des informations utiles sur son comportement dynamique.

La fonction STEP.

Elle se code usuellement

$$X = \text{STEP}(\text{HAUT}, \text{DEBUT})$$

d'où le symbole graphique donné ci-après



Si **TIME** est la variable temps dans le système modélisé, l'appelle de **STEP** correspond au sous-programme suivant donné en pseudo code

```

Si TIME < DEBUT
  Alors
    X = 0
  Sinon
    X = HAUT
Fin Si

```

D'où le graphe de principe



TSSD47 : Graphe de principe de la fonction step.

La fonction RAMP. Elle se code

```

X = RAMP(PENTE,DEBUT)

```

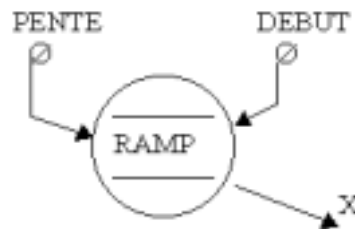
elle correspond au sous programme

```

Si TIME < DEBUT
  Alors
    X = 0
  Sinon
    X = PENTE*(TIME-DEBUT)
Fin Si

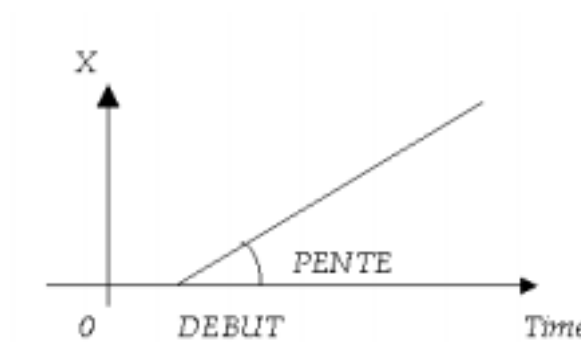
```

On lui associe usuellement le symbole graphique suivant



TSSD48

et le graphe de principe

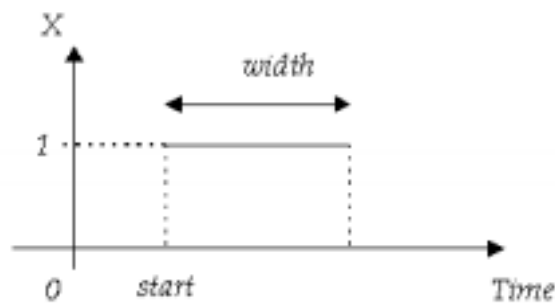


TSSD49

La fonctions PULSE. C'est une variante de la fonction impulsion de Dirac. Elle se code sous la forme

$$X = \text{PULSE}(\text{start}, \text{width})$$

Elle retourne la valeur 1 à partir de l'instant *start*, cette valeur perdure sur l'intervalle de longueur *width*. Autrement la valeur retournée est égale à 0. D'où le graphe de principe



TSSD50

Les fonctions sinusoïdales. Ce sont les fonctions *sin* et *cos*. En général on se sert de ces fonctions pour engendrer une **fonction sinusoïdale d'amplitude unité et de période donnée**. Rappelons que de façon générale une fonction sinusoïdale du temps est de la forme

$$y = f(t) = A \cdot \cos(\omega t - \varphi) \quad A > 0, \omega > 0$$

avec

A = amplitude
 $\omega t - \varphi$ = phase à l'instant t
 $-\varphi$ = phase en 0
 ω = pulsation
Période = $T = \frac{2\pi}{\omega}$ (> 0 par convention)
Fréquence = $1/T$

Dans le cas présent on se fixe la période comme $T = 2\pi/\omega$ on en déduit que $\omega = 2\pi/T$. Ainsi dans le langage Dynamo les fonctions *sin* et *cos* correspondent à

$$\sin\left(\frac{2\pi}{T} \cdot t\right) \quad \cos\left(\frac{2\pi}{T} \cdot t\right)$$

Notons que l'on peut généraliser ces fonctions jusqu'à obtenir des fonctions sinusoïdales amorties ou explosives avec seuil de la forme

$$y = f(t) = A \cdot e^{\alpha t} \cdot \cos(\omega t + \varphi) + S$$

explosive si $\alpha > 0$, amortie sinon

3.1.4.1.4 Les générateurs de bruit. Il servent à "bruiter", c'est à dire à apporter des éléments aléatoires dans l'information disponible dans un système modélisé. Il sont à utiliser avec parcimonie. Un modèle comprenant de tels générateurs n'est pas reproductible à l'identique compte tenu du caractère aléatoire des variables employées.

La fonction NOISE. En Dynamo elle se code `NOISE()`, les deux parenthèses servent à montrer qu'il s'agit d'une fonction, sans argument en l'occurrence. Elle sert à engendrer des nombres au hasard uniformément distribués entre -0.5 et +0.5, si on utilise un programme BASIC il s'agit de la fonction `RND(x)`, laquelle engendre des nombres compris entre 0 et 1.

La fonction NORMRN. Elle sert à engendrer des nombres au hasard distribués normalement avec une moyenne et un écart type spécifié. Elle se code dans DYNAMO avec deux arguments,

$$NORMRN(MOYEN, ECART)$$

où MOYEN = valeur moyenne désirée, ECART = écart type désiré. Il faut faire attention lorsque l'on utilise ce type de fonction. De fait soit $[a, b]$ le champ de variation désiré de notre variable. Si on note m la moyenne on a

$$m = \frac{a + b}{2}$$

Dans notre champ de variation on doit trouver 100% des observations. Or on sait que si

$$x \xrightarrow{\mathcal{L}} \mathcal{N}(m, \sigma) \Rightarrow \Pr(m - 3\sigma \leq x \leq m + 3\sigma) = 0.9973$$

Dès lors on peut prendre

$$\sigma = \frac{\frac{(b-a)}{2}}{3}$$

Mais dans ce cas, par le biais du générateur de nombres aléatoires, on risque de faire apparaître des aberrants situés en dehors du champ désiré. Par exemple si on se fixe le champ (0,10) alors on prendra $m = 5$ et $\sigma = 1.667$.

3.1.4.1.5 Fonctions logiques.

Fonctions MAX et MIN. Ce sont les fonction MAX(P,Q) et MIN(P,Q). La première donne la valeur la plus forte de P et Q, la seconde la valeur la plus faible de P et Q.

La fonction SAMPLE. Dans DYNAMO c'est une fonction à trois arguments qui permet d'échantillonner des valeurs d'une variable X à des intervalles réguliers dans le temps. Elle se code en langage DYNAMO

$$Y = \text{SAMPLE}(X, \text{INIT}, \text{INTVL})$$

avec

X : variable intrant à échantillonner
 INIT : valeur initiale de X
 INTVL : intervalle de temps d'échantillonnage.

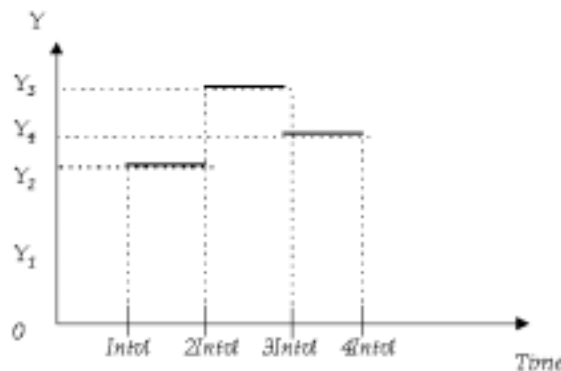
Cette fonction est telle que l'on peut lui associer le programme suivant dans lequel INT désigne la partie entière d'une variable.

```

Si TIME < INTVL
  Alors
    Y = INIT
  Sinon
    Fin si
Si TIME -(INTVL*INT(TIME/INTVL)) = 0
  Alors
    Y = X
  Sinon
    Fin Si

```

En d'autres termes chaque fois que le temps du modèle simulé est un multiple de INTVL, supérieur à 1, Y est égale à la valeur de X à cet instant. Elle conserve cette valeur jusqu'au prochain instant multiple de INTVL. D'où le graphe de principe : donné ci-après



TSSD51 : Graphe de principe de la fonction Sample.

La fonction CLIP. Elle permet de choisir entre deux quantités P et Q en fonction d'un critère CRIT comparé à une valeur référence REF. En Dynamo elle se code

$X = \text{CLIP}(P, Q, \text{CRIT}, \text{REF})$

Elle est telle que

```

Si CRIT < REF
  Alors
    X = Q
  Sinon
    X = P
  Fin Si

```

La fonction SWITCH. Elle permet elle aussi de faire un choix entre deux quantités P et Q, mais dans ce cas il n'y a qu'un troisième argument CRIT qui est ou non différent de zéro. Elle se code en Dynamo

$$X = \text{SWITCH}(P, Q, \text{CRIT})$$

Elle est telle que

Si $\text{CRIT} = 0$
 Alors
 $X = P$
 Sinon
 $X = Q$
 Fin Si

3.1.4.2 Les fonctions contenant des intégrations.

Ces fonctions servent d'une façon générale à simuler des décalages de temps, que ce soit dans des canaux de flux physique ou dans des chaînes d'information.

Chez Forrester on trouve fondamentalement 3 types d'assemblages (fonctions)

- Les DELAY [fonction avec ordre = nombre de niveaux]
- Les DLINF [fonction avec ordre]
- Les SMOOTH [fonction sans ordre]

Un DELAY est un assemblage de niveaux (variables d'état) et de flux (variables d'action) insérés dans un canal de flux physique de façon à modéliser la *transmission graduel d'un flux physique* d'un point d'entrée à un point de sortie.

DLINF est le pendant de DELAY mais cet assemblage agit dans un réseau d'information. Il est utilisé pour *créer un délai de transmission d'une information*. Si l'on considère une variable d'état Y, ou une variable auxiliaire A, on utilise DLINF pour engendrer une valeur perçue de Y, ou de A, notée VPY, ou VPA.

A la différence de DELAY et de DLINF, SMOOTH est une **fonction sans ordre**. Elle sert à engendrer la **moyenne exponentielle de premier ordre d'un flux physique**. C'est une **fonction de lissage exponentiel**. Comme DLINF cette fonction est insérée dans un réseau d'information

Si l'on se place à l'ordre 1, on montre que les 3 fonctions DELAY1, DLINF1 et SMOOTH se ramènent fondamentalement à la même équation différentielle (équation de niveau). C'est pourquoi les exégètes de Forrester confondent souvent les trois fonctions. On peut d'abord régler ce cas

3.1.4.2.1 Le cas des assemblages d'ordre 1. *****

Assemblage DELAY1.L'assemblage.

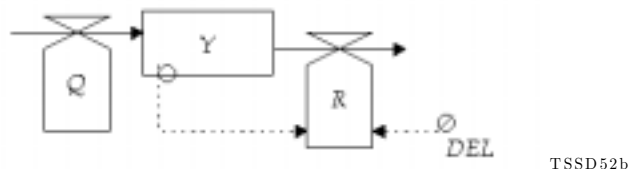
C'est l'assemblage fondamental en SSD. En Dynamo il se code

$$R = \text{DELAY1}(Q, \text{DEL})$$

où

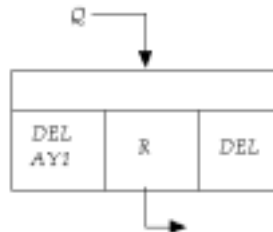
$$\begin{aligned} R &= \text{flux de sortie} \\ Q &= \text{flux d'entrée} \\ \text{DEL} &= \text{délai de temps} \end{aligned}$$

A cet assemblage on fait correspondre le diagramme de flux suivants



Diagrammes de flux de DELAY1.

Ou bien le symbole graphique agrégé



TSSD53

Lorsque le “compilateur” DYNAMO rencontre un ordre DELAY1, il engendre automatiquement l'ensemble d'équations suivant

$$\begin{aligned} Y &= Y + DT*(Q-R) & (L) \\ Y &= Y_0 & (N) \\ R &= Y/\text{DEL} & (R) \\ \text{DEL} &= D & (N) \end{aligned}$$

Dans **Principes des systèmes** Forrester propose une initialisation bien précise pour Y . Nous reviendrons ultérieurement sur ce problème. De plus

les niveaux et flux créés pour les besoins de la cause sont précédés de la lettre \$, ici on aurait par exemple $Y = \$Y$ ou $\$L$. Cet ensemble d'équations peut se ramener à une équation de niveau "réduite"

$$Y = Y + DT * (Q - Y/D)$$

Mais, *ce qui importe, c'est la sortie de l'assemblage* DELAY1, i.e. le flux R , or on sait que

$$R = \frac{Y}{DEL} \text{ ou } Y = DEL * R$$

d'où après remplacement de Y par cette valeur et simplification par DEL , l'équation de niveau donnant l'extrant de DELAY1

$$R = R + (DT/DEL) * (Q - R)$$

Il faut noter que Y peut fort bien ne pas avoir de correspondant matérialisable dans le monde réel, tout en se rapportant à un concept ou à un processus réels ayant une signification certaine. Par exemple Y peut représenter des "marchandises en commandes", c'est à dire des marchandises parties de chez le fournisseur et en cours d'expédition vers l'acheteur.

La signification de DEL.

On peut donner plusieurs interprétation au délai de temps DEL.

→ 1. C'est tout d'abord le **temps moyen** pendant lequel une unité de flux physique reste dans le niveau Y .

→ 2. C'est aussi le **temps nécessaire** pour que l'action induite par le flux Q produise un résultat, le flux R . En ce sens DEL est un **délai de transmission ou de transition**. On pourrait lui opposer les délais D que l'on retrouve dans les équations de flux usuelles, lesquels pourraient s'interpréter comme des **délais de politique, ou délais d'action**. A noter que chez certain auteurs DEL est appelé "magnitude" du délai.

Cela dit la fonction DELAY1 n'est qu'une fonction continue discrétisée (équation différentielle transformée) pour les besoins d'un processus de résolution numérique. On peut avoir une fonction discrète pure qui retourne la valeur de l'input décalé du délai de temps.

Le comportement analytique.

On part de l'équation de niveau réduite et l'on fait tendre DT (pas de calcul fini) vers dt (différentielle de t , ou intervalle de temps infiniment petit). On obtient une équation différentielle linéaire du premier ordre, à coefficients constants, non homogène

$$Y' + Y/D = Q(t)$$

En pratique $Q(t)$ est un flux dépendant d'un niveau (autre que Y) ayant un certain comportement temporel $f(t)$, c'est à dire que très généralement $Q(t) = g[f(t)]$. Ici, pour faciliter les choses nous prendrons $Q(t) = Q \forall t$. La solution de l'équation différentielle en Y est alors

$$Y(t) = Q \cdot D + (Y_0 - Q \cdot D) \cdot e^{-t/D}$$

Soit, en termes de réponse d'un DELAY1 à un intrant constant Q , et si on note X la réponse

$$X(t) = Q + \left(\frac{Y_0}{D} - Q \right) \cdot e^{-t/D}$$

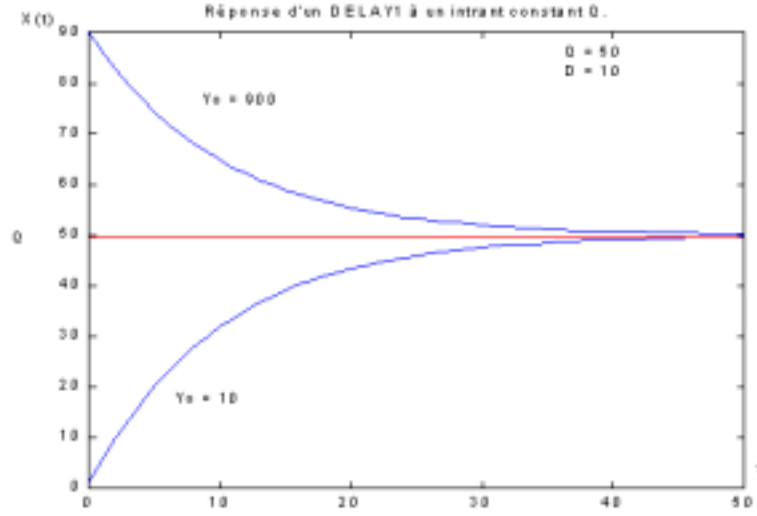
On peut alors faire plusieurs remarques

→ 1. Lorsque t tend vers l'infini, l'exponentielle tend vers zéro et donc X tend vers Q . C'est à dire que, **au cas d'un intrant constant dans un DELAY1, à terme, la sortie du DELAY1 est égale à cet intrant.**

→ 2. Si, initialement, $Y_0 = Q \cdot D$ et Forrester préconise cette initialisation dans son ouvrage Principes des Systèmes, il est alors bien évident que $X(t) = Q$ (ou $Y(t) = Q \cdot D$) et ce $\forall t$. C'est à dire que Q représente la **“valeur de régime permanent” ou “valeur steady-state”** de l'extrant de DELAY1. Dès lors si, dès le départ, le niveau constitutif du DELAY1 est initialisé de sorte qu'on soit en régime permanent on y reste ! Nous reviendrons par la suite sur ce problème des initialisations préconisées par Forrester.

→ 3. La forme de l'équation solution obtenue (en Y ou en X) permet de justifier l'appellation de **“ Délai exponentiel de premier ordre ”** donnée à DELAY1. *Exponentiel* compte tenu de la forme de la fonction, *de premier ordre* parce que dans la boucle de rétroaction liant le niveau Y au flux de sortie, via le délai D , on ne rencontre qu'un seul niveau, qu'une seule variable d'état. Rappelons que l'ordre d'un modèle de SSD est égal au nombre de variables d'état mises en jeu.

Le comportement temporel de $X(t)$ est décrit dans la figure suivante



TSSD54b : Réponse d'un DELAY1 à un intrant constant Q .

Au vu de cette figure on note que

→ 1 Selon la valeur d'initialisation on converge vers la valeur d'état stationnaire Q , par valeurs supérieures ou inférieures.

→ 2 Si on trace la tangente à l'origine de la courbe analytique, celle-ci coupe la droite horizontale d'ordonnée Q à l'instant $t = D$ (délai de temps du DELAY1). De fait la pente de la courbe $X(t)$ est donnée par sa dérivée soit

$$X'(t) = -\frac{1}{D} \cdot \left(\frac{Y_0}{D} - Q \right) \cdot e^{-t/D}$$

D'où en $t = 0$

$$X'(0) = -\frac{1}{D} \cdot \left(\frac{Y_0}{D} - Q \right)$$

Comme $X(0) = Y_0/D$, l'équation de la tangente à l'origine est

$$y = \frac{Y_0}{D} - \frac{1}{D} \cdot \left(\frac{Y_0}{D} - Q \right) \cdot t$$

Si l'on fait $y = Q$ il vient

$$\begin{aligned} \frac{Y_0}{D} - \frac{1}{D} \cdot \left(\frac{Y_0}{D} - Q \right) \cdot t &= Q \Leftrightarrow \frac{Y_0}{D} - Q = \frac{1}{D} \cdot \left(\frac{Y_0}{D} - Q \right) \cdot t \\ \Leftrightarrow 1 &= \frac{t}{D} \Leftrightarrow t = D(cqfd) \end{aligned}$$

On retrouve de fait le comportement d'une boucle de rétroaction négative du premier ordre à seuil positif. Dans le cas présent le seuil est égal à l'intrant Q .

Cas de DLINF1.

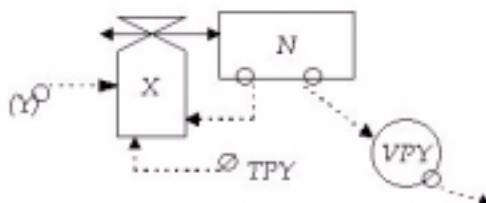
Si l'on considère une variable d'état Y , ou une variable auxiliaire A , on utilise DLINF1 pour engendrer une valeur perçue de Y , ou de A , notée VPY , ou VPA . En Dynamo il se code

$\begin{aligned} VPY &= DLINF1(Y,TPY) \text{ ou} \\ VPA &= DLINF1(A,TPA) \end{aligned}$

où

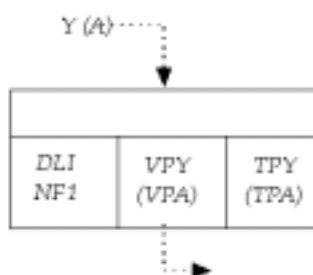
$\begin{aligned} VP &= \text{valeur de sortie} \\ Y \text{ ou } A &= \text{variable observée} \\ TPY \text{ ou } TPA &= \text{temps de perception de} \\ &\quad \text{la vraie valeur (temps de lissage)}. \end{aligned}$

A cet assemblage on fait correspondre le diagramme de flux donné ci-après.



TSSD60b : Diagrammes de flux de DLINF1.

Ou bien le symbole graphique agrégé



TSSD61

NB : Contrairement à DELAY1, les flèches associées à ce symbole sont en tirets. Cela est dû au fait que la fonction est insérée dans un réseau d'information.

Lorsque le compilateur DYNAMO rencontre l'ordre DLINF1, il engendre automatiquement un ensemble d'équations. Nous l'écrirons dans le cas d'une

variable d'état Y (les équations seraient similaires pour une auxiliaire A).

$$\begin{array}{l} X = (Y - N)/TPY \quad (R) \\ N = N + DT*X \quad (L) \\ N = Y \quad (N) \\ VPY = N \quad (A) \end{array}$$

La double flèche associé au flux X dans le diagramme de flux usuel donné ci-dessus, tient au fait que selon les valeurs relatives de Y et N ce flux peut être positif ou négatif. Cet ensemble d'équations peut se ramener à une "équation de niveau réduite"

$$\begin{aligned} VPY &= VPY + (DT/TPY)*(Y - VPY) \\ \Leftrightarrow V &= V + (DT/D)*(Y - V) \end{aligned}$$

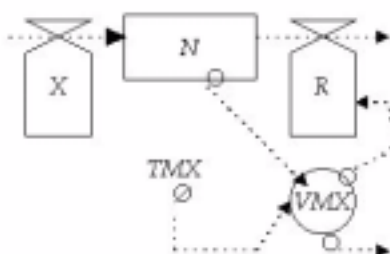
On retrouve aux notations près l'équation de niveau de DELAY1.

L'assemblage SMOOTH.

Comme DLINF cette fonction est insérée dans un réseau d'information. Considérons un flux X dont on désire calculer la valeur moyenne VMX , sur une certaine période de temps, notée TMX (pour Temps de lissage Moyen de X). La fonction se code alors

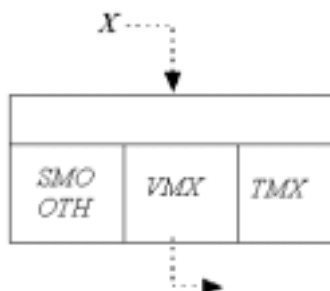
$$VMX = \text{SMOOTH}(X, TMX)$$

A cette fonction on fait correspondre le diagrammes de flux



TSSD65 : Diagramme de flux de SMOOTH.

Ou bien on peut utiliser un symbole graphique agrégé



Remarque : Comme dans le cas de DLINF on notera que les flèches d'arrivée et de sortie du niveau (dans le diagramme développé, comme dans le symbole agrégé) sont en tirets. Ici encore on travaille sur de l'information et non sur des flux physiques.

Quand il rencontre l'ordre SMOOTH le compilateur Dynamo engendre la série d'équations suivantes

$$\begin{array}{l} N = N + DT^*(X - R) \quad (L) \\ N = X^*TMX \quad (N) \\ VMX = N/TMX \quad (A) \\ R = VMX \quad (R) \end{array}$$

Cet ensemble d'équations peut encore se résumer par une "équation de niveau réduite", si l'on note $VMX = V$ et $TMX = D$, il vient

$$V = V + (DT/D)^*(X - V)$$

En faisant tendre DT vers dt , infiniment petit, on obtient une équation différentielle du premier ordre qui est formellement identique à celle obtenue au cas de DLINF1 ou de DELAY1. De fait on a

$$dV = \frac{dt}{D} \cdot (X - V) \Leftrightarrow \dot{V}(t) + \frac{V(t)}{D} = \frac{X(t)}{D}$$

Rapport avec les formules usuelles de lissage exponentiel.

Si l'on considère une série de valeurs numériques temporelles on peut vouloir en faire la moyenne sur une certaine période, en lissant ces valeurs. Une solution simple est la moyenne mobile. Par, exemple si l'on a des observations mensuelles, on fait la moyenne des trois observations du premier trimestre, cette moyenne est rapportée au mois médian. Puis on se décale d'un mois, on fait la moyenne sur trois mois, on la rapporte au mois médian. On poursuit ainsi le processus. D'où le schéma de principe donné ci-après.

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \left. \begin{array}{l}] \\] \\ \} \\] \\] \end{array} \right\} \rightarrow m_1 \quad \left. \begin{array}{l}] \\] \\ \} \\] \\] \end{array} \right\} \rightarrow m_2 \quad \left. \begin{array}{l}] \\] \\ \} \\] \\] \end{array} \right\} \rightarrow m_3$$

On note que dans cette procédure on affecte le même poids à chacune des observations. Or bien souvent, dans la réalité, plus on s'éloigne dans le temps plus l'information tend à "s'affaiblir". D'où l'idée qui est à la base du lissage exponentiel, donner une valeur d'autant plus faible à l'information que l'on s'éloigne dans le temps. Pour ce faire on peut adopter des coefficients de pondération en progression géométrique, $\lambda, \lambda^2, \lambda^3, \dots, \lambda^n, \dots$ ($0 < \lambda < 1$).

On suppose que l'on observe les valeurs temporelles à des intervalles réguliers de valeur dt , si l'on note " $-i$ " l'indice à l'instant " $-i \cdot dt$ ", la valeur lissée en 0 est égale à

$$L_0 = \frac{X_{-1} + \lambda \cdot X_{-2} + \dots + \lambda^n \cdot X_{-(n+1)} + \dots}{1 + \lambda + \lambda^2 + \dots + \lambda^n + \dots}$$

Or on sait que

$$1 + \lambda + \dots + \lambda^n = \frac{1 - \lambda^{n+1}}{1 - \lambda} \xrightarrow{n \rightarrow +\infty} \frac{1}{1 - \lambda}$$

Ainsi l'on a

$$L_0 = (1 - \lambda) \cdot [X_{-1} + \lambda \cdot X_{-2} + \dots + \lambda^n \cdot X_{-(n+1)} + \dots]$$

De même en $0 + dt$ il vient

$$L_{dt} = (1 - \lambda) \cdot [X_0 + \lambda \cdot X_{-1} + \dots + \lambda^n \cdot X_{-n} + \dots]$$

soit

$$L_{dt} = (1 - \lambda) \cdot X_0 + \lambda \cdot L_0$$

Le coefficient λ représente le poids de l'information en $-dt$ par rapport à l'information en 0, c'est donc une fonction de dt . On peut alors poser, $\lambda = 1 - (dt/T)$, où T représente une période globale d'observation, un temps global d'observation pendant lequel on réalise le lissage. Il vient alors

$$L_{dt} = \frac{dt}{T} \cdot X_0 + \left(1 - \frac{dt}{T}\right) \cdot L_0 = L_0 + \frac{dt}{T}(X_0 - L_0)$$

On retrouve bien ainsi la formule de lissage qui est à la base des fonctions utilisées en SSD.

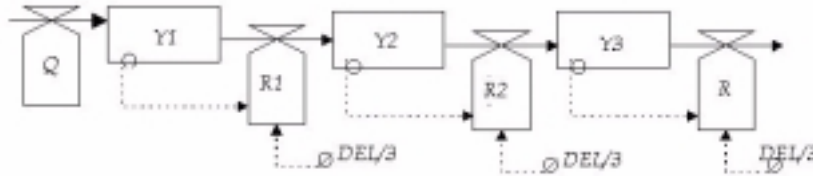
3.1.4.2.2 Cas des assemblages d'ordre supérieur à 1. *****

Délais exponentiels d'ordre supérieur à 1, DELAY3, DELAYn.

Un délai exponentiel d'ordre n est un assemblage de n DELAY1 mis "en série", ou "en cascade". Des considérations purement empiriques ont conduit à privilégier l'assemblage DELAY3 comme suffisamment représentatif des phénomènes de transition de flux physiques observés dans la réalité. C'est celui que l'on emploie le plus souvent pour représenter des DELAY généraux.

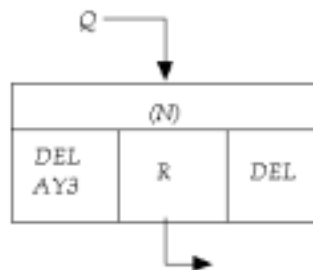
L'assemblage DELAY3.

Son diagramme de flux est une démultiplication de celui de DELAY1, d'où le diagramme



TSSD55 : Diagramme de flux de DELAY3.

On peut ici encore utiliser un symbole graphique agrégé



TSSD56

On notera qu'à la différence de DELAY1 on a fait figurer ici une lettre dans le bandeau supérieur du symbole. De fait, au vu de la littérature existante en SSD, on peut distinguer **trois types de fonctions**

→ 1. La fonction DELAY3, simple dans laquelle le flux Q est transformé en flux R , sans plus.

→ 2. La fonction DELAY3P (P pour "Pipe") dans laquelle, non seulement le flux Q est transformé en R , mais de plus on donne un nom à la somme des trois niveaux constitutifs du DELAY3. (Cf., par exemple, Exercice 13, Future Electronics Model in Michael R. GOODMAN Study Notes in System Dynamics, Cambridge Mass., Wright Allen Press 1974 pp. 349-364)

→ 3. Une variante mixte des deux précédentes : on donne un nom à un niveau qui a pour flux d'entrée Q , et pour flux de sortie R obtenu par un DELAY3, l'équation de niveau s'écrivant alors de façon usuelle, $N = N + DT(Q - R)$. (Cf., par exemple, Exercice 11, Analysis of Market Growth Model in Michael R. GOODMAN Study Notes in System Dynamics, Cambridge Mass., Wright Allen Press 1974 pp. 281-308).

On a ainsi deux types d'appels de fonctions dans un compilateur Dynamo

$$R = \text{DELAY3}(Q, \text{DEL}) \quad (1)$$

où

Q = flux d'entrée,
 DEL = délai de temps affecté au lissage

$$\boxed{R = \text{DELAY3P}(Q, \text{DEL}, N)} \quad (2)$$

où

Q = flux d'entrée,
 DEL = délai de temps affecté au lissage
 N = nom du niveau égal à la somme des
 3 niveaux constitutifs de DELAY3.

Lorsque le compilateur DYNAMO rencontre l'ordre DELAY3 il engendre automatiquement l'ensemble d'équations donné ci après

$$\begin{aligned} \$L1 &= \$L1 + DT*(Q - R1) \quad (L) \\ \$L1 &= Q*(DEL/3) \quad (N) \\ \$R1 &= \$L1/(DEL/3) \quad (R) \\ \$L2 &= \$L2 + DT*($R1 - $R2) \quad (L) \\ \$L2 &= Q*(DEL/3) \quad (N) \\ \$R2 &= \$L2/(DEL/3) \quad (R) \\ \$L3 &= \$L3 + DT*($R2 - R) \quad (L) \\ \$L3 &= Q*(DEL/3) \quad (N) \\ R &= \$L3/(DEL/3) \quad (R) \end{aligned}$$

NB

1. On notera que chaque niveau constitutif est initialisé à $Q*(DEL/3)$, ce qui suppose que l'on soit à l'état stationnaire (Cf. infra pour l'explication).

2. Le délai de transmission total du flux intrant est égal à DEL , comme chacun des DELAY1 constitutifs est équivalent aux deux autres, on affecte à chacun de ces DELAY1 un délai égal au tiers du délai total.

Comportement analytique de DELAY3, cas d'une entrée constante.

En prenant comme base les équations engendrées par le compilateur Dynamo on peut se ramener à 3 équations de niveau réduites, dans ce qui suit on remplacera $\$Li$ par Y_i , on notera X_i les flux construits $\$Ri$ et on prendra $DEL = D$

$$\begin{aligned} Y_1 &= Y_1 + DT \times \left[Q - \frac{Y_1}{D/3} \right] \\ Y_2 &= Y_2 + DT \times \left[\frac{Y_1}{D/3} - \frac{Y_2}{D/3} \right] \\ Y_3 &= Y_3 + DT \times \left[\frac{Y_2}{D/3} - \frac{Y_3}{D/3} \right] \end{aligned}$$

Ces équations peuvent être transformées en équations différentielles en vue de la recherche d'une solution analytique.

$$Y_1' + \frac{3Y_1}{D} = Q \Leftrightarrow Y_1' + kY_1 = Q \quad (1)$$

$$Y_2' + \frac{3Y_2}{D} = \frac{3Y_1}{D} \Leftrightarrow Y_2' + kY_2 = kY_1 \quad (2)$$

$$Y_3' + \frac{3Y_3}{D} = \frac{3Y_2}{D} \Leftrightarrow Y_3' + kY_3 = kY_2 \quad (3)$$

La résolution de ces trois équations se fait “en cascade” à partir du moment où l’on connaît la forme de la fonction intrant $Q(t)$. La méthode utilisée pour chaque équation est celle déjà employée, solution générale sans second membre, variation de la constante et retour aux conditions initiales pour déterminer la constante d’intégration. Comme précédemment, dans le cas de DELAY1, on peut poser $Q(t) = Q\forall t$, et initialiser les trois niveaux constitutifs à la valeur 0. On a alors les solutions suivantes

$$\begin{aligned} Y_1(t) &= \left(\frac{QD}{3}\right) \cdot (1 - e^{-3t/D}) \\ Y_2(t) &= Q \cdot \left[\frac{D}{3} - \left(t + \frac{D}{3}\right) \cdot e^{-3t/D}\right] \\ Y_3(t) &= Q \cdot \left[\frac{D}{3} - \left(\frac{3t^2}{2D} + t + \frac{D}{3}\right) \cdot e^{-3t/D}\right] \end{aligned}$$

Dans la mesure où l’on s’intéresse à l’extrait du DELAY3 il faut avoir les équations de comportement des flux et non celles des niveaux constitutifs. Donc il faut diviser chacune des équations précédentes par $D/3 = (1/k)$, soit

$$\begin{aligned} X_1(t) &= Q \cdot (1 - e^{-3t/D}) \\ X_2(t) &= Q \cdot \left[1 - \left(\frac{3t}{D} + 1\right) \cdot e^{-3t/D}\right] \\ X_3(t) &= Q \cdot \left[1 - \left(\frac{9t^2}{2D^2} + \frac{3t}{D} + 1\right) \cdot e^{-3t/D}\right] \end{aligned}$$

sachant que de fait $X3(t) = R(t)$!

NB Pour avoir le cas particulier de la réponse d’un DELAY1 il suffit de prendre l’équation de $X1(t)$ et de remplacer $D/3$ par D . D’où la **réponse d’un DELAY1**

$$R(t) = Q \cdot (1 - e^{-t/D})$$

De même, dans le cas d’un DELAY2, il suffit de prendre l’équation de $X2(t)$ et de remplacer $D/3$ par $D/2$. D’où la **réponse d’un DELAY2**

$$R(t) = Q \cdot \left[1 - \left(\frac{2t}{D} + 1\right) \cdot e^{-2t/D}\right]$$

Si on initialise les 3 niveaux constitutifs à la valeur N (valeur quelconque) on note que la constante d’intégration est égale à

$$N - \frac{QD}{3} = N - \frac{Q}{k} = N - q$$

On a alors

$$\begin{aligned} X_1(t) &= kY_1(t) = Q + (kN - Q) \cdot e^{-3t/D} \\ X_2(t) &= kY_2(t) = Q + (kN - Q) \cdot [kt + 1] \cdot e^{-3t/D} \\ X_3(t) &= kY_3(t) = Q + (kN - Q) \cdot \left[\frac{k^2 t^2}{2} + kt + 1 \right] \cdot e^{-3t/D} \end{aligned}$$

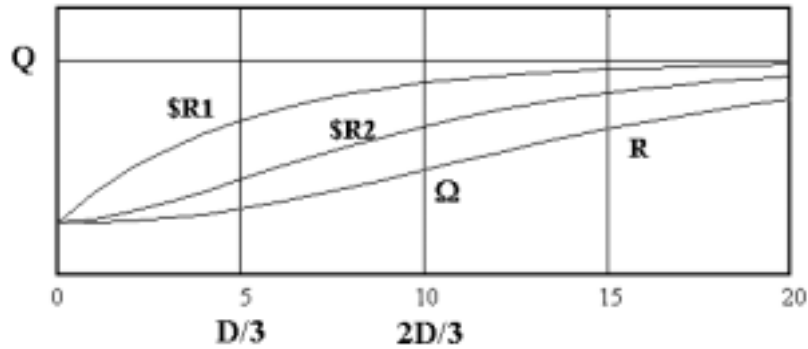
soit, pour ce qui est de l'extrait de DELAY3

$$X(t) = R(t) = \frac{Y_3(t)}{D/3} = Q + \left(\frac{3N}{D} - Q \right) \cdot \left[\frac{9}{D^2} \frac{t^2}{2} + \frac{3t}{D} + 1 \right] \cdot e^{-3t/D}$$

On note que lorsque t augmente indéfiniment les trois flux tendent vers Q – ou $Q+$ selon que l'on a $N < Q/k = QD/3$ ou $N > Q/k$. Par ailleurs si originellement $N = QD/3$, alors $X(t) = Q$ et ce quel que soit t . Encore une fois l'initialisation préconisée par J.W. Forrester suppose que l'on soit à l'état stationnaire et que de facto on y reste sauf modification exogène !

Puisque ce qui importe est la réponse de DELAY3 à un certain intrant, ici Q , on peut étudier l'allure des courbes de comportement temporel des flux. On supposera que les trois niveaux constitutifs sont initialisés à la valeur $N < Q/k = QD/3$.

L'évolution temporelle des trois flux de DELAY3, avec une entrée Q et des initialisations des niveaux constitutifs à N est donnée dans le diagramme ci-dessous.

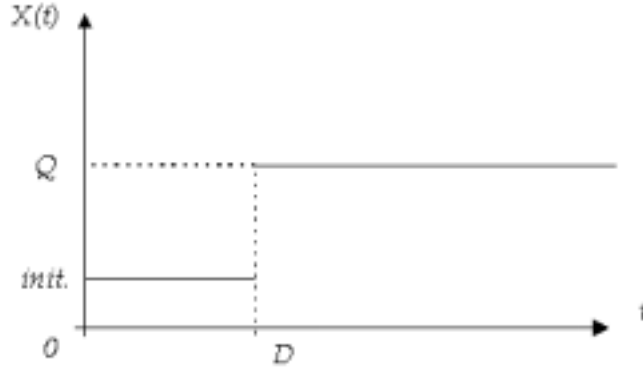


TSSD57 : Flux d'un DELAY3 en réponse à un intrant constant.

Remarque : Pour le flux final de sortie on note l'existence d'un point d'inflexion pour $t = 2D/3 \equiv (n-1)D/n$, où n représente l'ordre du DELAY. Alors

$$Si \quad n \rightarrow +\infty \Rightarrow \frac{n-1}{n}D \rightarrow D$$

En d'autres termes si l'on a un DELAY d'ordre infini, la courbe en S se transforme en une courbe "échelon" pure et le "saut" de la valeur initiale à la valeur d'état stationnaire Q se fait à l'instant $t = D$. D'où le graphe de principe donné ci-après



TSSD58 : Réponse d'un DELAY infini à un intrant constant.

De fait ce résultat est confirmé par la généralisation à un DELAY $_n$

La généralisation à DELAY $_n$.

On pose $n \in \mathbb{N} - \{1\}$, on suppose que le délai de transition totale est égal à D , et que l'on a $Q(t) = Q \ \forall t$, $Y_i(0) = N$. Par ailleurs on pose : $r = n/D$, $q = Q/r = QD/n$, alors il vient

$$\forall i \neq 1 \quad Y_i(t) = q + (N - q) \cdot e^{-rt} \cdot \left[\sum_{j=1}^{n-1} r^j \frac{t^j}{j} + 1 \right] \quad t \in \mathbb{R}^+$$

$$\text{Si } i = 1 \quad Y_1(t) = q + (N - q) \cdot e^{-rt} \quad \forall n \in \mathbb{N}$$

De plus, $\forall i \ X_i(t) = r \cdot Y_i(t)$, dès lors la réponse d'un DELAY $_n$, dans les conditions prises ici, est

$$R(t) = Q + \left(\frac{nN}{D} - Q \right) \cdot e^{-rt} \cdot \left[\sum_{j=1}^{n-1} r^j \frac{t^j}{j} + 1 \right]$$

Quand $n \rightarrow +\infty$, on a

$$e^{-rt} = \exp\left(-\frac{nt}{D}\right) \rightarrow 0^+$$

Alors $R(t) \rightarrow Q$ puisque l'exponentielle l'emporte sur toute fonction puissance.

Le problème de l'initialisation.

L'initialisation des niveaux constitutifs d'un DELAY_n selon la préconisation de J.W. Forrester ne semble se justifier que si l'on est déjà à l'état stationnaire (*Steady State*). De fait si l'on considère la plupart des modèles rencontrés dans la littérature SSD on note tout d'abord une distinction entre

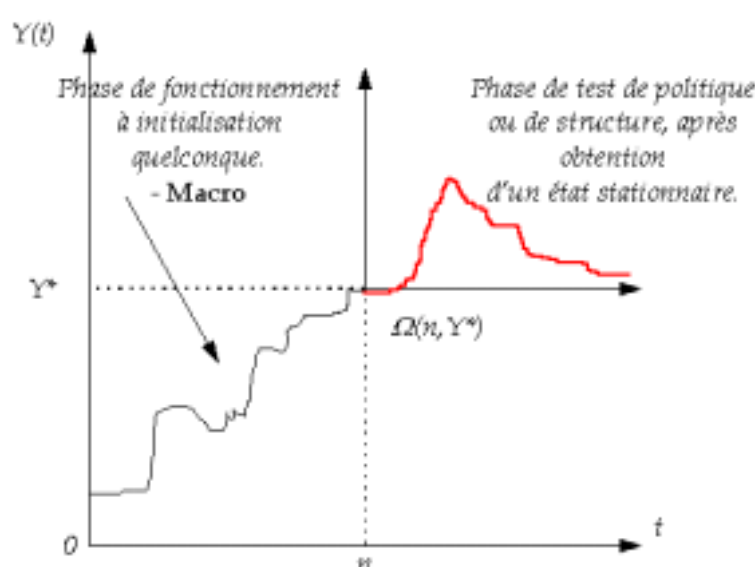
Modèles de type *macro-économique* (du genre “ *Dynamique mondiale* “ ou “ *Dynamique urbaine* “),

Modèles de type *micro-économique* (du genre “ *Future Electronics Model* “ in M.R. GOODMAN **Study Notes in System Dynamics** Wright Allen Press Cambridge Mass 1974 pp.349-364).

Dans ce second type de modèle, les valeurs initiales qui sont fournies sont des valeurs d'état stationnaire et l'on analyse la réponse du modèle à une excitation exogène, ou même on cherche à modifier les politiques mises en jeu. Tout se passe comme si l'on nous fournissait un modèle qui a déjà fonctionné avec des valeurs initiales quelconques afin d'atteindre un état d'équilibre.

Par contre, dans les modèles de type macro-économique, les valeurs initiales sont en général des valeurs d'une année de référence. On fait alors fonctionner le modèle jusqu'à obtention d'un état stationnaire et c'est seulement une fois que celui-ci est atteint que l'on cherche à tester des politiques.

On peut ainsi dresser un schéma de principe concernant l'évolution temporelle d'une variable d'état (Y) d'un modèle de SSD et les procédures sous-jacentes



TSSD59 : Schéma de principe de la distinction micro / macro en SSD.

Compte tenu de ce que nous venons de dire on sait qu'il y a une formule commune aux trois assemblages de “premier ordre” (SMOOTH est

sans ordre tout en incorporant une boucle de premier ordre). Si l'on a une formule unique la variable "Entrée" est pour sa part multiforme. Dans le cas de DELAY1 c'est un flux. Dans le cas de DLINF1 c'est un niveau ou une auxiliaire, dans le cas de SMOOTH c'est à nouveau un flux. D'où les **problèmes d'initialisation**

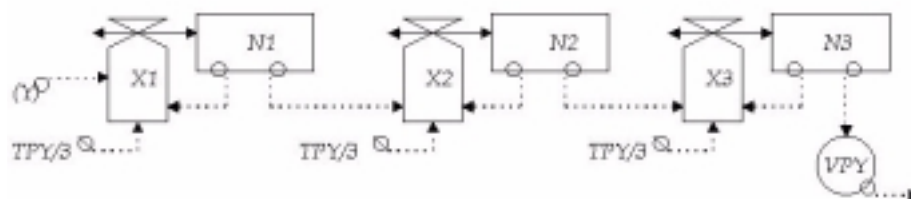
- Pour DELAY1 valeur initiale du flux d'entrée (problème).
- Pour DLINF1 valeur initiale du niveau d'entrée (sans problème) ou de l'auxiliaire d'entrée (problème).
- Pour SMOOTH valeur initiale du flux d'entrée (problème).

Assemblage DLINF3.

Comme dans le cas de DELAY3, DLINF3 est un assemblage de trois DLINF1 en "cascade". Si on prend le cas d'une variable d'état Y il se code en Dynamo

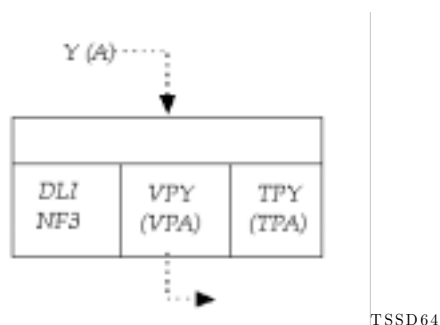
$$VPY = DLINF3(Y,TPY)$$

Le diagramme de flux développé de cette fonction est le suivant (bien noter les différences avec DELAY3 !)



TSSD63 : Diagramme de flux de DLINF3.

On peut aussi utiliser un symbole graphique agrégé



TSSD64

Lorsque le compilateur DYNAMO rencontre la fonction DLINF3 il engendre automatiquement l'ensemble d'équations suivant

$ \begin{aligned} X1 &= (Y - N1)/(TPY/3) & (R) \\ N1 &= N1 + DT \cdot X1 & (L) \\ N1 &= Y & (N) \\ X2 &= (N1 - N2)/(TPY/3) & (R) \\ N2 &= N2 + DT \cdot X2 & (L) \\ N2 &= Y & (N) \\ X3 &= (N2 - N3)/(TPY/3) & (R) \\ N3 &= N3 + DT \cdot X3 & (L) \\ N3 &= Y & (N) \\ VPY &= N3 & (A) \end{aligned} $
--

Ici encore on peut se ramener à 3 équations de niveau réduites, soit en notant $TPY = D$

$$\begin{aligned}
N_1 &= N_1 + \frac{DT}{D/3} \cdot [Y - N_1] \\
N_2 &= N_2 + \frac{DT}{D/3} \cdot [N_1 - N_2] \\
N_3 &= N_3 + \frac{DT}{D/3} \cdot [N_2 - N_3]
\end{aligned}$$

En faisant tendre DT vers dt on se ramène à 3 équations différentielles

$$\begin{aligned}
N_1' + \frac{3N_1}{D} &= \frac{3Y}{D} \\
N_2' + \frac{3N_2}{D} &= \frac{3N_1}{D} \\
N_3' + \frac{3N_3}{D} &= \frac{3N_2}{D}
\end{aligned}$$

On peut résoudre ce système en cascade à partir du moment où l'on connaît la forme de l'intrant $Y(t)$. Comme dans le cas de DLINF1 on posera $Y(t) = at$. On note que chaque niveau constitutif est alors initialisé à 0. La réponse à cet intrant est donnée par la valeur de N_3 , et l'on a

$$VPY(t) = N_3(t) = a \cdot (t - D) + a \cdot \left(\frac{3}{D} \cdot \frac{t^2}{2} + 2t + D \right) \cdot e^{-3t/D}$$

Remarques.

→ 1. On notera que cette équation représente, dans les mêmes conditions, la réponse d'un DELAY3 à un flux d'entrée d'équation $Q(t) = at$. Cela est bien l'indice d'une similitude entre les assemblages.

→ 2. De même que pour DLINF1, lorsque t augmente indéfiniment, le terme complémentaire fonction d'une exponentielle à exposant négatif tend vers zéro. Donc, à terme, la valeur perçue de Y , à un instant donné t , est égale à la vraie valeur de Y à l'instant $(t - D)$. Il y a bien, à terme, un décalage de perception de valeur $D = TPY$.

3.1.4.3 Passage du continu discrétisé au discret en SSD. Fonctions de type “ PIPE ”.

Ainsi que nous l'avons souligné en présentant les fonctions DELAY, il existe une différence notable entre le continu discrétisé pour les besoins d'un processus de résolution numérique et le discret proprement dit. Comme nous le montrons par ailleurs (PSB, Référentiels de temps et évolution dynamique - Problèmes de méthode, **Revue Internationale de Systémique**) on peut utiliser une fonction purement discrète, appelée “ PIPE ” de façon générique, pour simuler la transmission discrète de flux.

Nous employons le terme “ PIPE ” par référence aux pipelines dans lesquels on peut faire passer des liquides différents, grâce à des tampons intermédiaires, ce afin de conserver l'idée de “ transmission par paquets ”. Cette fonction PIPE possède un ordre. On la notera généralement PIPE_n . On a ainsi n “ boîtes ” en série. Le flux intrant à l'instant t dans la boîte 1, sortira à l'identique à l'instant $t + D$ (D = délai du PIPE) de la boîte n . Si l'on conserve les notations déjà employées, cette fonction peut se coder

$$\boxed{FS = \text{PIPE}_n(FE, D)}$$

Ce codage est quelque peu redondant puisque n , donnant l'ordre du PIPE, est tel que l'on ait de facto $n \equiv D$. Nous supposons ainsi que le temps de transition d'une quantité physique dans PIPE est un nombre entier (D). On peut symboliser graphiquement cette fonction comme suit



TSSD66 : Schéma de principe de la fonction PIPE.

Le schéma peut s'interpréter de la façon suivante. Une quantité physique précise rentrant à la date t dans B_1 [noté encore $B(1)$], soit FE , ressortira de B_n à l'identique, sous la forme de FS , en $t + D$. Si l'on note $FB(i)$ le contenu de la boîte $B(i)$ qui à un moment donné va passer dans la boîte $B(i + 1)$, on a, à un instant donné, pour un flux d'entrée FE donné

$$\boxed{\begin{array}{l} \forall i \in (1, n) : FB(i) = B(i) \\ FS = FB(n) \end{array}}$$

Et le processus de transition se traduit par

$$\boxed{\begin{array}{l} B(1) = B(1) + FE - FB(1) \\ \forall i \in (2, n) : B(i) = B(i) + FB(i - 1) - FB(i) \end{array}}$$

A noter qu'initialement toutes les boîtes $B(i)$ ont pour valeur 0.

Telle quelle cette fonction peut remplacer dans la méthodologie Dynamique de Systèmes forresterienne les variables de niveaux et les fonctions de lissage exponentiel contenant une intégration comme DELAYn (pour DLINFn et SMOOTH on utilisera DELAY INFORMATION). C'est une fonction discrète en ce sens que le passage d'une boîte $B(i)$ à une boîte $B(i+1)$ ne peut se faire qu'à des instants entiers du temps. Cette fonction échappe ainsi au DT , le pas d'intégration de la Dynamique de Système " usuelle ". Ou plus précisément le DT du processus de résolution numérique ne peut être que l'unité de temps du système modélisé. Le flux FS acquiert ainsi un caractère discret. En effet il est soumis à une condition du type

'' FS à l'instant t est égal à FE à l'instant t - D ''

Par contre, le flux FE conserve son caractère continu usuel de la Dynamique de Systèmes. Il n'est pas soumis à une condition du genre : " FE joue si t à telle valeur ". C'est pourquoi nous pouvons parlé de Simulation de Systèmes Dynamiques en " temps mixte ". A noter qu'il existe des logiciels comme "EXTEND" qui peuvent réaliser des simulations continues ou discrètes pures. Ceci est bien évidemment possible avec MATLAB et SIMULINK.

3.1.5 Addenda : cartes de contrôle, tracés de courbes et impressions numériques sous Dynamo.

Afin de faciliter la lecture des programmes SSD, tirés de la littérature courante en ce domaine, nous donnons ci-après des explications concernant les "cartes de contrôle", les ordres d'impression graphique ou numérique ainsi que la lecture des graphes issus du compilateur Dynamo.

3.1.5.1 "Control cards" ou "Control Statements".

Nous donnons ci-après des exemples courants des "cartes de contrôle" se trouvant à la fin des programmes Dynamo usuels. Celles ci décrivent les paramètres principaux d'une simulation ainsi que les ordres d'impression.

Paramètres de simulation. Ceux ci sont essentiellement le pas de calcul et la longueur du processus. Le codage usuel est le suivant

C DT = 1 → Pas de calcul = 1 unité de temps
 C LENGTH = 250 → Durée du processus = 250 unités de temps

Période de traçage des courbes (PLTPER). Nous donnons ci-après des exemples d'ordres indiquant la périodicité du traçage des courbes issues d'une simulation.

C PLTPER = 2 $\rightarrow$ Le tracé se fait toutes les 2 unités de temps

ou (Cf. Urban Dynamics)

```
A PLTPER.K = CLIP(PLTMIN,PLTMAX,PLTCT,TIME.K)
C PLTMIN = 5
C PLTMAX = 5
C PLTCT = 500
```

$\Rightarrow PLTPER = PLTMIN$ si $PLTCT \geq TIME.K$
 $PLTPER = PLTMAX$ si $PLTCT < TIME.K$
 On trace en fait tous les 5 ans quel que soit $TIME$

ou (Cf. Tsembaga2)

```
A PLTPER.K=CLIP(PLTPR2,PLTPR1,TIME.K,SWT)
C PLTPR1 = 10
C PLTPR2 = 1
C SWT = 100
```

$\Rightarrow PLTPER = PLTPR2$ si $TIME.K \geq SWT$
 $PLTPER = PLTPR1$ si $TIME.K < SWT$
 En dessous de 100 on trace tous les 10 ans,
 au dessus tous les ans.

Période d'impression des résultats (PRTPER). On trouvera ci-après un exemple d'ordres donnant la périodicité d'impression des résultats numériques d'une simulation. Cet exemple est issu du modèle "Dynamique urbaine".

```
A PRTPER.K = CLIP(PRTMIN,PRTMAX,PRTCT,TIME.K)
C PRTMIN = 50
C PRTMAX = 100
C PRTCT = 160
```

$\Rightarrow PRTPER = PRTMIN$ si $PRTCT \geq TIME.K$
 $PRTPER = PRTMAX$ si $PRTCT < TIME.K$
 Avant 160 on imprime tous les 50 ans
 au delà tous les 100 ans.

Ordre de tracé des courbes. Il s'agit de l'ordre PLOT. Nous en donnons deux exemples.

```
(Urban Dynamics)
PLOT U=U,PH=P,WH=W(0,600E3)/NE=E,DI=D(0,40E3)
```

Cet ordre signifie que l'on trace U avec le symbole U , PH avec le symbole P , WH avec le symbole W et les trois avec une échelle allant de 0 à 600.000. De même on trace NE avec le symbole E , DI avec le symbole D et les deux avec une échelle allant de 0 à 40.000.

```
(Tsembaga)
PLOT HP=H(200,300)/PP=P(0,200)/YPA=*(3E6,5E6)/I=N(0,.1)
```

Ici on trace HP avec le symbole H et une échelle allant de 200 à 300, on trace PP avec le symbole P et une échelle allant de 0 à 200, YPA est tracé avec le symbole $*$ et une échelle allant de 3 millions à 6 millions, enfin I est tracé avec le symbole N et une échelle allant de 0 à 0.1.

NB.

→ 1- Tous les graphes issus de Dynamo sont toujours tracés avec un axe vertical divisé en 4 parties égales.

→ 2- Le caractère “/” sert à séparer les groupes de symboles ayant une même échelle, le caractère “,” sert à séparer les variables avec leur affectation symbolique de tracé.

Parmi les modèles distribués au titre du dossier dans le DESS on note les caractéristiques suivantes Pour le modèle “Société Précision” il n’y a pas de cartes de contrôle. Une indication d’échelle pour tous les tracés graphiques dans les modèles, “Communauté résidentielle”, “Croissance de marché”, “Eutrophication des lacs”, “Tsembaga”. Il manque des échelles dans les ordres PLOT pour les modèles “Plateau du Kaïbab” et “Future Electronics”.

Ordre d’impression des résultats numériques.2 Il s’agit de l’ordre PRINT. On ne le rencontre que dans le modèle “Dynamique urbaine” le seul pour lequel sont donnés des résultats numériques. Dans tous les autres modèles rencontrés ne figurent que des résultats graphiques. Dans ce modèle les impressions numériques se font par “blocs”. la syntaxe est la suivante.

```
PRINT 1) (3.0) AV
PRINT 1) (3.3) DI,DID,DIDP,L(3.2)
...
PRINT 9) (0.3) UMM,UMP,UR,WHAM
```

Ce qui se lit comme suit. Imprimer en colonne 1 avec le format (3.0) la variable *AV*. Puis imprimer, à la suite, en colonne 1 avec le format (3.3) les variables *DI*, *DID*, *DIDP*, avec le format (3.2) la variable *L*.

Enfin imprimer en colonne 9 avec le format (0.3) les variables *UMM*, *UMP*, *UR*, *WHAM*. Toute impression d'une nouvelle variable dans une même colonne fait passer à la ligne suivante.

Pour ce qui est des formats leur lecture se fait comme suit :

(3.0) = variable en milliers avec aucun chiffre après le point décimal,

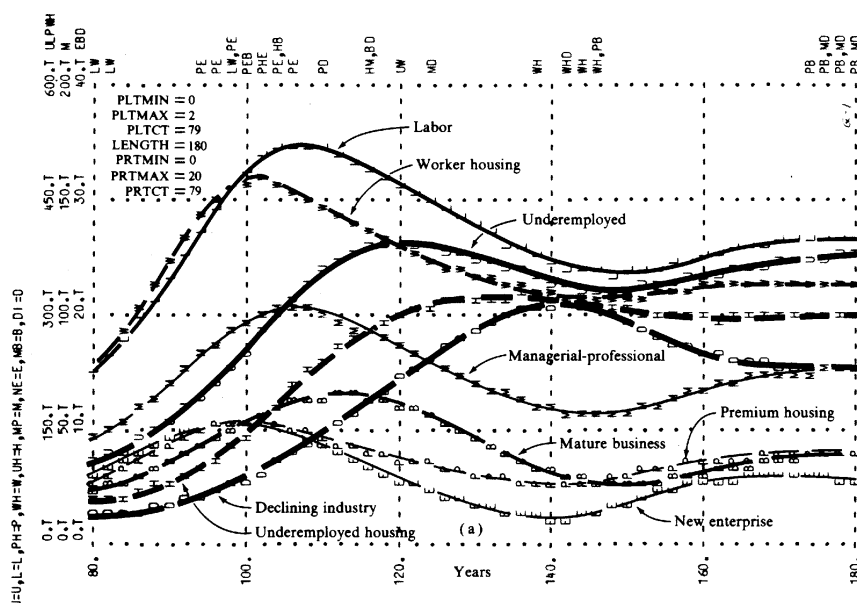
(3.2) = variable en milliers avec 2 chiffres après le point décimal,

(0.3) = variable en unités avec 3 chiffres après le point décimal.

3.1.5.2 Lecture des tracés de courbes temporelles.

Les graphiques issus du compilateur Dynamo donnent le comportement temporel de certaines variables sélectionnées. L'axe horizontal donne le temps, dans l'unité de temps adoptée pour le modèle. Dans la marge de gauche on trouve, à l'extrême gauche, une ligne verticale donnant les symboles utilisés pour le tracé de chaque variable sélectionnée. Par exemple, dans le graphique présenté ci-après $U = U$ signifie que la variable *U* est tracée à l'aide du symbole *U*.

Les **échelles en ordonnée** sont données sur la **gauche du graphe**. On trouve à leur extrémité supérieure les symboles de tracé auxquels elles s'appliquent. Par exemple la première échelle allant de 0 à 600.000 (la lettre **T** suivant chaque valeur de l'échelle numérique signifie milliers, "**thousands**") s'applique aux variables tracées avec les symboles, *U*, *L*, *P*, *W* et *H*. Tout le long du sommet du graphe les groupes de lettres indique l'endroit où les courbes se chevauchent. Par exemple, dans la figure, à l'instant 145 les lettres *WH,PB* signifient que les courbes de symboles *W* et *H* se coupent et le premier symbole *W* est le seul tracé, de même les courbes de symboles *P* et *B* se coupent aussi et sont alors représentées par le premier symbole *P*.



Vsm034 : Un tracé usuel du logiciel "Dynamo".

3.2 Future Electronics et Simulink.

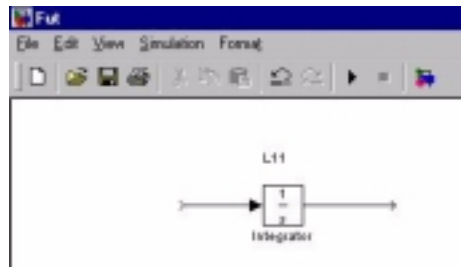
3.2.1 La transcription du modèle.

Munis des éléments fondamentaux on peut maintenant traduire le modèle "Future Electronics" sous Simulink. Pour ce faire on utilise l'énoncé donné en annexe et notamment, dans cet énoncé, le premier diagramme de flux, ainsi que les équations.

On ouvre MATLAB, puis SIMULINK. On ouvre un nouveau modèle. On le nommera FUTELEC, il sera stocké, par défaut dans le répertoire bin du dossier MATLAB.

On a une structure de boucles de rétroaction, mais il nous faut bien commencer à un endroit. On va d'abord s'occuper de la transformation du flux TE en flux TFF via un $DELAY3P$ donnant le niveau $TEFO$.

Les 3 niveaux constitutifs seront notés L11, L21 et L31, ce sont 3 intégrateurs de Simulink. On commence par amener un intégrateur sur notre feuille blanche. On lui adjoint une flèche d'entrée (quantité à intégrer) et une flèche de sortie (quantité intégrée). L'intégrateur se nomme L11.



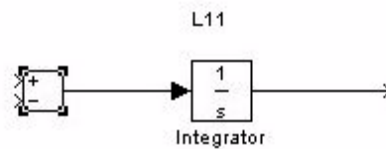
FE001

Il faut faire attention à ne pas se laisser tromper par la symbolique graphique. Dans un diagramme de flux à la JWF un niveau (rectangle) est alimenté par un flux d'entrée (TE) et vidé par un flux de sortie (TFF). Sous Simulink le flux net à intégrer figure en entrée d'un bloc Intégrateur. Ici ce flux net est constitué par

- Le "flux d'entrée" : TE (signe +)
- Le "flux de sortie" : $R11 = \frac{L11}{DDF/3}$ (signe -)

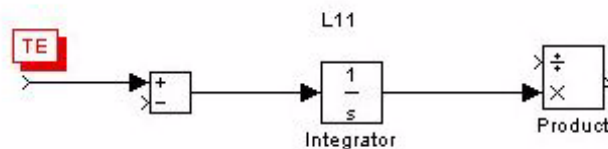
Il nous faut donc porter sur la flèche d'entré un bloc "Sum" dont les paramètres seront :

- Icon shape : rectangular
- List of signs : + -



FE002

L'entrée (+) de ce bloc est constituée par TE , l'entrée (-) par $L11/(DDF/3)$. La quantité $L11$ sort du bloc intégrateur elle est à diviser par $(DDF/3)$. On porte donc sur la flèche de sortie un bloc "product" de paramètres / * (ils correspondent aux entrées dans l'ordre vertical).



FE003

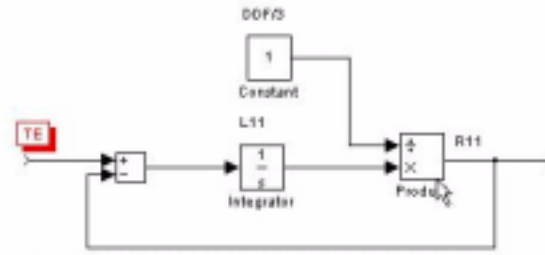
L'entrée ($\times$) de ce bloc est constituée par $L11$, l'entrée ($\div$) par $DDF/3$. Pour s'éviter un problème de rajout de bloc intermédiaire on crée un input constante égal à

$$\frac{DDF}{3} = \frac{3}{3} = 1$$

qui va constituer l'entée ($\div$) du bloc "Product". La sortie de ce bloc n'est autre que

$$R11 = \frac{L11}{DDF/3}$$

On la récupère comme entrée (-) du bloc "Sum" et elle va servir d'entrée (+) pour le prochain bloc intégrateur donnant $L21$.



FE004

On réitère les opérations en ajoutant deux intégrateurs donnant $L21$ et $L31$. En fait on sélectionne les 3 blocs principaux (Sum, Integrator et Product) et on fait un double copier coller, en raboutant correctement les flèches et en modifiant les noms de façon adéquate. On n'oubliera pas notamment que la sortie du troisième intégrateur donne le flux TFF .

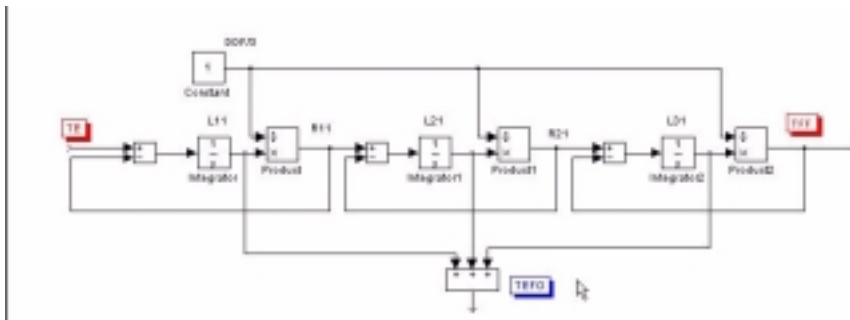
On notera que, au fur et à mesure des adjonctions, on se déplace vers la droite de l'écran, un "ascenseur" horizontal faisant son apparition. On peut toujours essayer de resserer le dessin par des déplacements idoines.

Puisque l'on a affaire à un $DELAY3P$ la somme des 3 variables d'état constitutives est égale à $TEFO$

$$L11 + L21 + L31 = TEFO$$

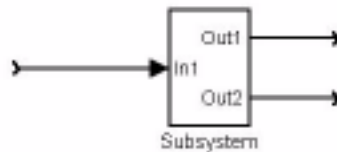
$\Rightarrow$ adjonction d'un bloc "Sum" de forme rectangulaire avec 3 entrées (+ + +) dont la sortie donne $TEFO$. (On fait tourner ce bloc de façon adéquate à l'aide de Ctrl + R).

On obtient ainsi l'assemblage suivant



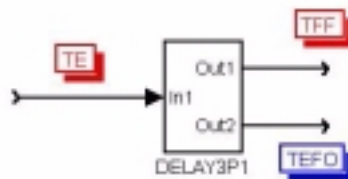
FE005

Cet assemblage est quelque peu encombrant. Pour régler le problème on va créer un sous système. On sélectionne tous nos blocs (avec la souris ou en faisant *Edit + Select All = Ctrl+A*) puis on fait *Edit & Create Subsystem* ou *Ctrl + G*. On se retrouve avec un bloc ayant une entrée (*In1*) et deux sorties (*Out1* et *Out2*). Il faut réarranger les flèches pour compacter les choses.



FE006

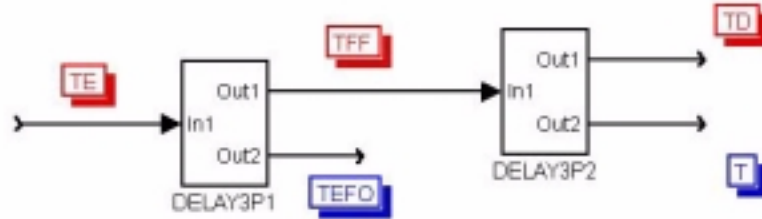
Un double clic gauche sur la boîte Subsystem fait apparaître une fenêtre *Futelec/Subsystem** qui permet de voir que $In1 = TE$, $Out1 = TFF$ et $Out2 = TFEFO$, ces entrées et sorties figurant dans des cartouches. On peut reporter ces noms aux places adéquates dans notre schéma. On en profite pour cliquer sur Subsystem et on change l'appellation en *DELAY3P1*.



FE007

On peut alors continuer notre progression dans le diagramme de flux. Le flux *TFF* entre dans un nouveau *DELAY3P* pour donner le flux *TD*, sachant que les 3 niveaux constitutifs donnent la variable *T*. L'assemblage *DELAY3P*, *per se*, étant déjà constitué notre tâche est simplifiée. On copie

le bloc *DELAY3P1* et on fait un collage (oh miracle le nom se transforme en *DELAY3P2*). L'entrée de ce sous-système n'est autre que *TFF* et l'on a $Out1 = TD$ et $Out2 = T$. On place les flèche et les noms de façon adéquate (pour les noms on copie, colle et transforme *TFF* et *TEFO*).

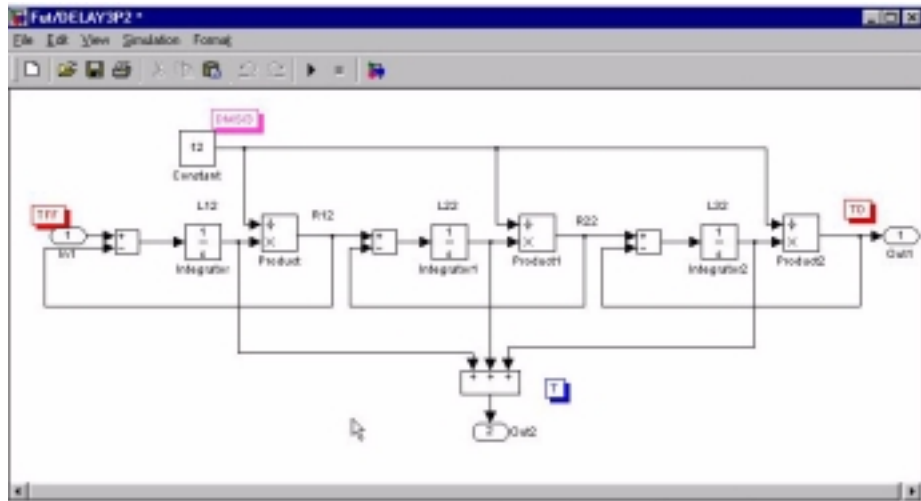


FE008

On fait un double clic gauche sur *DELAY3P2* de façon à modifier les noms pour cet assemblage. On fait cela à titre de pure "documentation" du modèle, car les noms que l'on donne n'ont aucune influence sur les calculs effectués par Simulink. Par contre il est nécessaire de modifier la valeur de la constante délai de temps dans cet assemblage puisque désormais on a

$$\frac{DMS}{3} = \frac{36}{3} = 12$$

On doit obtenir un sous-système qui a l'allure suivante.

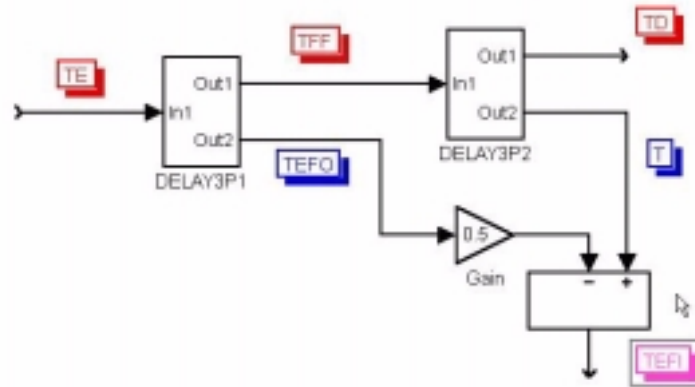


FE009

On traduit maintenant l'équation

$$TEFI = T - 0.5 * TEFO$$

c'est l'équation (4,A) de l'énoncé dans laquelle on a supprimé les "post-indices", ceux-ci n'ayant pour nous qu'une valeur "historique". Il nous faut effectuer une sommation généralisée (introduction d'un bloc Sum) en faisant le produit $0.5 * TEFO$, on utilise pour ce faire un bloc "Gain". Il vient

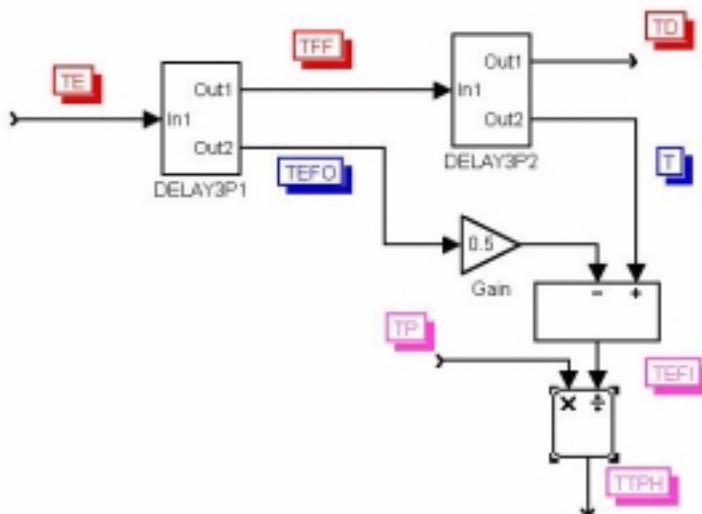


FE010

On peut maintenant traduire "partiellement" l'équation (5,A)

$$TTPH = \frac{TP}{TEFI}$$

nous disons partiellement puisque TP est inclus dans une boucle démarrante en $TTPH$ pour y revenir. On introduit ainsi un bloc Product, dont on cache le nom (*Clic droit + Format + Hide Name*), son entrée ($\div$) est constituée par $TEFI$, de son entrée ($\times$) on fait partir une flèche nommée TP , la sortie du bloc donne $TTPH$. On a ainsi



FE011

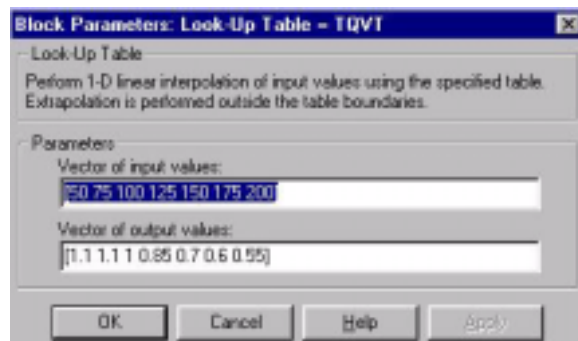
L'auxiliaire $TTPH$ sert d'intrant pour une Table donnant QR . On doit ainsi traduire les équations

$$\begin{aligned} QR &= TABLE(TQVT, TTPH, 50, 200, 25) \quad (6, A) \\ TQVT &= 1.1/1.1/1/0.85/0.7/0.6/0.55 \quad (6.1, T) \end{aligned}$$

Pour ce faire on utilise un bloc *Look Up Table* (Librairie Simulink, Functions & Tables, Look Up Table). Lorsque l'on sélectionne ce bloc on a le commentaire suivant qui apparaît

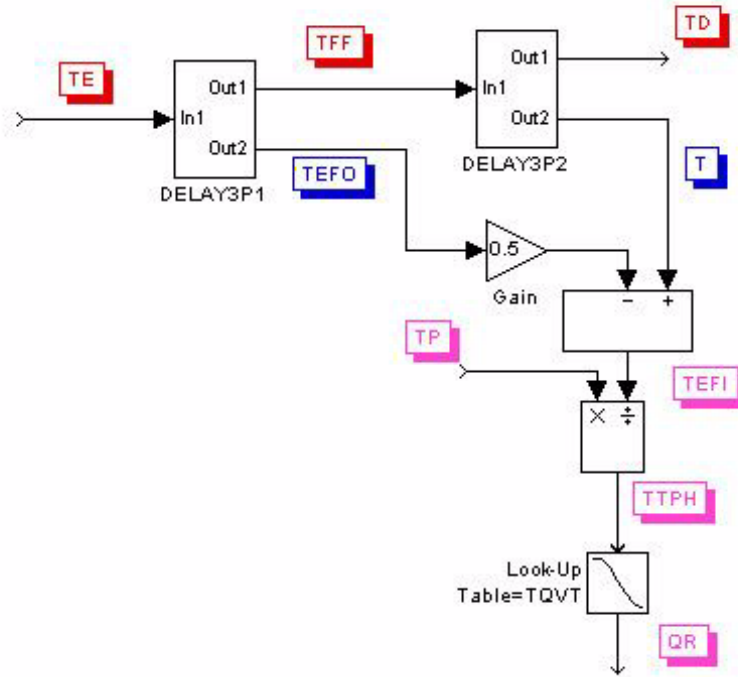
Perform 1-D linear interpolation of input values using the specified table. Extrapolation is performed outside the table boundaries.

En d'autres termes ce bloc se comporte comme la fonction TABHL de DYNAMO ! L'entrée du bloc est constituée par $TTPH$ (faire subir une rotation au bloc pour que l'entrée corresponde). On complète le nom du bloc en écrivant *Look-Up Table = TQVT*. On effectue un clic droit et on prend *Block Parameters* pour décrire les deux vecteurs d'entrée et de sortie, soit



FE012

On notera les entrées sous forme vectorielle et surtout le fait que lorsque l'on clique sur OK on voit apparaître la forme graphique de la fonction dans le bloc ! La sortie de la table est égale à QR . Il vient



FE013

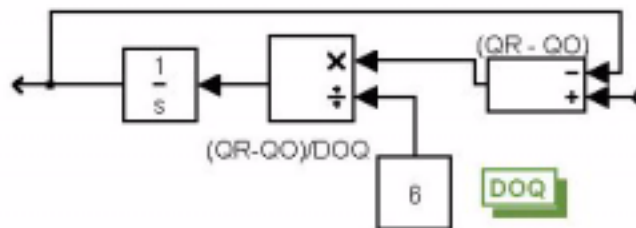
Dans l'énoncé original cette auxiliaire QR alimente une fonction SMOOTH pour donner l'auxiliaire QO . On notera le fait que c'est un SMOOTH qui est employé plutôt qu'un DLINF1. Cela étant on va traduire cette fonction SMOOTH sous forme d'un sous système. En fait on traduit l'équation DYNAMO

$$QO = QO + \left(\frac{DT}{DOQ} \right) \times (QR - QO)$$

c'est à dire que, à l'aide d'un bloc "Integrator" on intègre la quantité

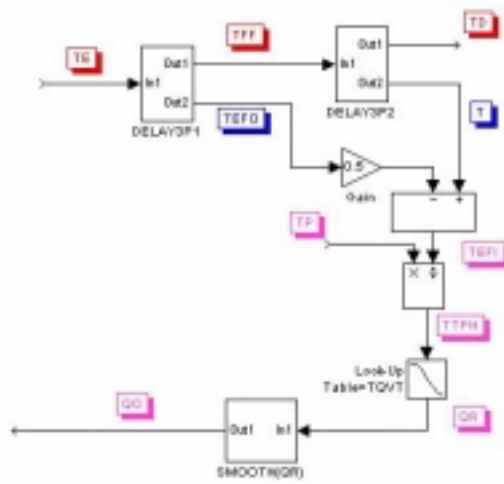
$$\frac{QR - QO}{DOQ}$$

sachant que la sortie de ce bloc donne la quantité QO . On a ainsi l'assemblage suivant



FE014

que l'on met sous forme de sous-système noté $\text{SMOOTH}(\text{QR})$, dont l'entrée (In1) est égale à QR et la sortie (Out1) à QO . D'où le schéma global

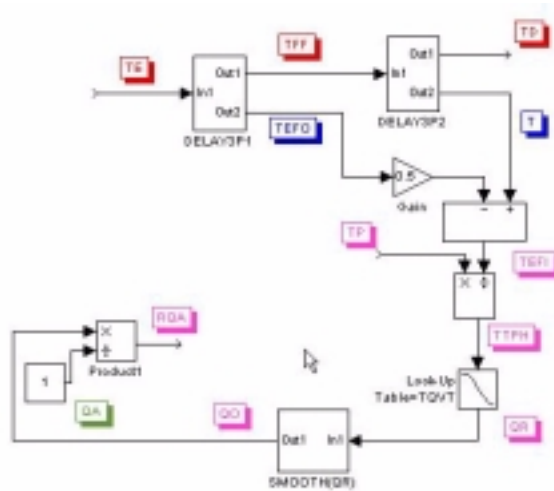


FE015

L'auxiliaire QO sert à créer RQA , on a

$$\begin{aligned} \text{RQA} &= \text{QO}/\text{QA} \quad (8, A) \\ \text{QA} &= 1 \quad (8.1, C) \end{aligned}$$

On introduit une source "Constant" qui donne QA et on réalise l'opération QO/QA avec un bloc Product. Soit



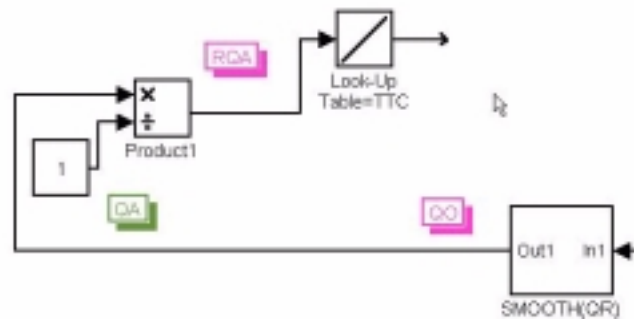
FE016

A partir de RQA deux boucles démarrent. L'une revient vers $TTPH$, l'autre va vers TE . Occupons nous de la première. On rencontre en premier la table donnant TC . En fait on a dans l'énoncé

$$TC = TABLE(TTC, RQA, 0, 1.5, 0.5) * (1 + STEP) \quad (12, A)$$

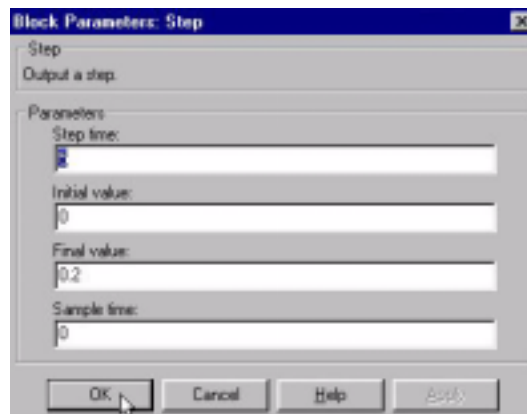
$$TTC = 0/500/1000/1500 \quad (12.1, T)$$

La fonction STEP permet d'avoir une excitation exogène, si cette fonction n'agit pas le système doit rester en état stable (*steady state*). Commençons par introduire une *Lok Up Table* pour TTC .



FE017

Il reste à réaliser l'opération $TTC * (1 + STEP)$ qui exige une addition et un produit. La fonction STEP est disponible sous SIMULINK, dans la librairie *Sources*. On introduit un bloc Step et par un double clic gauche, on active le panneau Block Parameters que l'on remplit comme suit



FE018

Puis on traduit l'opération $(1 + STEP)$ avec un bloc *Constant* et un bloc *Sum* et on réalise un produit pour avoir TC . Il vient globalement


$$X1 = \frac{TC - N1}{DDP/3}$$

Du point de vue de la différence de logique avec le DELAY3 on note que le bloc Product s'insère entre le bloc Sum et le bloc Integrator, alors qu'auparavant il était en sortie du bloc Integrator. En réitérant les opérations on a l'assemblage global.