# Résumé de LEPL1106

compilation du 20 février 2023

Thomas Debelle

Juin 2023

# Table des matières

# Préface

Bonjour à toi !

Cette synthèse recueille toutes les informations importantes données au cours, pendant les séances de tp et est amélioré grâce au note du Syllabus. Elle ne remplace pas le cours donc écoutez bien les conseils et potentielles astuces que les professeurs peuvent vous donner. Notre synthèse est plus une aide qui on l'espère vous sera à toutes et tous utiles.

Elle a été réalisée par toutes les personnes que tu vois mentionné. Si jamais cette synthèse a une faute, manque de précision, typo ou n'est pas à jour par rapport à la matière actuelle ou bien que tu veux simplement contribuer en y apportant ta connaissance ? Rien de plus simple ! Améliore la en te rendant ici où tu trouveras toutes les infos pour mettre ce document à jour. (*en plus tu auras ton nom en gros ici et sur la page du github*)

Nous espérons que cette synthèse te sera utile d'une quelconque manière ! Bonne lecture et bonne étude.

# Première partie

# Signaux

# Chapitre 1

# Les signaux

## 1.1 Définition

#### Définition

Un signal est une fonction de une ou plusieurs variables (continues ou discrètes) qui correspondent à de l'information ou a un phénomène physique.

#### Continues ou discrets ?

Un signal est dit continu si il est définit sur un espace de temps continu. On note ce signal $x(t)$. Et il est dit discret si il est définit sur un espace discret de temps. On note ce signal $x[t]$.

#### Manipulation des signaux

Pour le cas discrets ou continu, nous pouvons réaliser les opérations suivantes.

— Combinaison linéaire $\rightarrow \alpha_1 x_1(t) + \alpha_2 x_2(t)$
— Multiplication $\rightarrow x_1[t]x_2[t]$
— Dilatation $\rightarrow x[n/a], a > \mathbb{R}$
— Translation $\rightarrow x(t - t_0), t_0 \in \mathbb{R}$
— Renversement $\rightarrow x(-t)$
— Différentiation (que pour le cas continu) $\rightarrow \frac{d^n x(t)}{dt^n}$
— Intégration (que pour le cas continu)

## 1.2 Signaux élémentaires

### 1.2.1 Signaux exponentiels

Pour les signaux continus nous avons :

$$x(t) = Be^{at} \tag{1.1}$$

Et pour les signaux discrets nous avons :

$$x[n] = Br^n \rightarrow 0 < r < 1 \tag{1.2}$$

### 1.2.2 Signaux sinusoïdales

Pour les signaux continus nous avons :

$$x(t) = Acos(\omega t + \phi) \tag{1.3}$$

Et pour les signaux discrets nous avons :

$$x[n] = Acos(\Omega n + \Phi) \tag{1.4}$$

Il a une période de $\Omega N = 2\pi m$

### 1.2.3 Signaux amortis

Pour les signaux continus et avec $\alpha > 0$ :

$$x(t) = Ae^{-\alpha t}cos(\omega t + \phi) \tag{1.5}$$

Et pour les signaux discrets :

$$x[n] = Br^n cos(\Omega n + \Phi) \tag{1.6}$$

### 1.2.4 L'impulsion (temps discret)

Comme son nom l'indique, ce signal se représente sous la forme d'une impulsion. Par sa définition, cela nous force a avoir un signal discret !

$$\begin{cases} \delta[n] = 1 \rightarrow n = 0 \\ \delta[n] = 0 \rightarrow \forall n \notin 0 \end{cases} \tag{1.7}$$

On peut réaliser des impulsions décaler en écrivant $\delta[n-x]$ avec $x$ représentant la valeur du décalage.

### 1.2.5 L'échelon (temps discret)

Ce type de signal élémentaire est encore plus trivial puisqu'il se résume à :

$$\begin{cases} 1 \rightarrow n \geq 0 \\ 0 \rightarrow n < 0 \end{cases} \tag{1.8}$$

On peut aussi voir l'échelon comme une somme infinie d'impulsion comme $\sum_{k\geq 0}^{\infty} \delta[n-k]$.
Il existe également un échelon en temps continu qui se résume à la même équation 1.8 mais pour des valeurs continues.

### 1.2.6 L'impulsion (temps continu)

$$\begin{cases} \delta(t) = 0 sit \neq 0 \\ \delta(0) = ?(+\infty) \\ \int_{-a}^{a} \delta(s)ds = 1 \rightarrow \forall a > 0 \end{cases} \tag{1.9}$$

A noter que la dernière ligne nous crée une propriété bien spécifique. En effet, la valeur de l'impulsion est limité par les bornes $a$ puisqu'on impose une intégrale égale à 1.

#### Lien entre impulsion et échelon

$\delta(t) = u'(t)$ donc l'impulsion est une sorte de dérivé de l'échelon. ceci nous permet également d'obtenir cette formule :

$$\int_{-\infty}^{\infty} x(s)\delta(s)ds = x(0) \tag{1.10}$$

On prouve cela de manière *peu rigoureuse* en remarquant que : pour $s \neq 0$, on a $\delta(s) = 0$ donc $x(s)\delta(s) = 0 = x(0)\delta(s)$ finalement on a :

$$\int_{-\infty}^{\infty} x(s)\delta(s)ds = \int_{-\infty}^{\infty} x(0)\delta(s)ds = x(0)\int_{-\infty}^{\infty} \delta(s)ds = x(0) \tag{1.11}$$

#### Décomposition en impulsions

On peut décomposer tout signal en impulsion comme ci-dessous :

$$\begin{cases} \text{temps discret } \Rightarrow x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \\ \text{temps continu } \Rightarrow x(t) = \int_{-\infty}^{infty} x(s)\delta(t-s)ds \end{cases} \tag{1.12}$$

## 1.3 Convolution

La convolution est une nouvel opérateur qui nous sera très utile. Son signe est "$*$" et la formule qui définit cette opération est :

$$f[n] * g[n] := \sum_{k=-\infty}^{\infty} f[k]g[n-k] \tag{1.13}$$

La méthode pour trouver le résultat d'une convolution est :

1. Il faut "*renverser*" une des fonctions. C'est-à-dire faire $f[n] \to f[-n]$.
2. Puis multiplier ces deux nouvelles fonctions.
3. L'*aire* sous la courbe créer par cela est le résultat de la convolution.

# Deuxième partie

# Systèmes

# Chapitre 2

# Système LIT

Un système est une entité qui prend *un ou plusieurs signaux* en entrée et produit *de nouveaux signaux* en sortie.
Exemple : $H\{x[n]\} = x[n] + x[n-1]$. Un système est par exemple : *une radio, une caméra, une voiture, ...*

## 2.1 LIT

Un système *Linéaire et Indépendant du Temps* ou **LIT** est, comme son nom l'indique, linéaire donc :

$$\mathcal{H}\{a_1x_1 + ... + a_Nx_N\} = a_1\mathcal{H}\{x_1\} + ... + a_N\mathcal{H}\{x_N\} \tag{2.1}$$

Et un système est *invariant temporelle* donc :

$$\begin{cases} \mathcal{H} \text{ est invariant si } \forall t_0 \in \mathbb{N} \\ \mathcal{H}\{x\}[n] = y[n] \Rightarrow \mathcal{H}\{x[n-n_0]\} = y[n-n_0] \end{cases} \tag{2.2}$$

On remarque qu'on peut ré-écrire tous signaux via une somme d'impulsions. De plus, si $\mathcal{H}$ est linéaire alors :

$$\mathcal{H}\{x\} = \mathcal{H}\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\} = \sum_{k=-\infty}^{\infty} x[k]\mathcal{H}\{\delta[n-k]\} \tag{2.3}$$

Si $\mathcal{H}$ est *invariant* au temps et qu'on pose $h := \mathcal{H}\{\delta\}$

$$\mathcal{H}\{\delta[n-k]\} = \mathcal{H}\{\delta\}[n-k] = h[n-k] \tag{2.4}$$

$$\mathcal{H}\{x\} = \sum_{k=-\infty}^{\infty} x[k]h[n-k] =: x * h \tag{2.5}$$

A noter que "$*$" fais référence à la convolution, sujet abordé à la section 1.3.

## 2.2 Opération sur les signaux

Voici un tableau résumant les différentes opérations possibles sur les signaux :

| | Temps discret | Temps continu |
|---|---|---|
| Combinaison linéaire | $\alpha_1 x_1[n] + \alpha_2 x_2[n]$ | $\alpha_1 x_1(t) + \alpha_2 x_2(t)$ |
| Multiplication | $x_1[n]x_2[n]$ | $x_1(t)x_2(t)$ |
| Différentiation | | $d^n x(t)/dt^n$ |
| Intégration | | $\int_{-\infty}^{t} x(\tau)d\tau$ |
| **Convolution** | $x_1[n] * x_2[n]$ | $x_1(t) * x_2(t)$ |
| Dilatation | $x[n/a]$ (arrondi $n/a$) | $x(t/a)a > 0$ |
| Translation | $x[n-n_0], n_0 \in \mathbb{Z}$ | $x(t-t_0), t_0 \in \mathbb{R}$ |
| Renversement | $x[-n]$ | $x(-t)$ |

Une chose importante à voir dans ces formules est que $x$ est un *signal*, $x[k]$ la *valeur* de ce signal en k et on a $\mathcal{H}$ qui est un *système* prenant et donnant des signaux.

## 2.3 Modélisation et représentation des systèmes

Comme vu précédemment, la réponse impulsionelle est le résultat du système étant perturbé par une impulsion.

### 2.3.1 Inconvénient

1. Fonction de taille *infinie* et représentation donc *peu simple.*
2. La modélisation d'un système ne mène généralement pas à une réponse impulsionelle .
3. On doit connaitre l'entrée depuis $-\infty$

### 2.3.2 Représentation

Il existe 3 grandes façons de représenter un système. Tout d'abord la méthode *équation différentielle d'entrée-sortie.* Pour la suite des exemples, j'utiliserai un circuit $RLC$ comme montré ci-contre.

*équation différentielle d'entrée-sortie* est une somme des dérivées comme montré dans l'équation 2.6. C'est plutôt facile de trouver les équations mais on fait face à un problème, l'opération *dérivée* n'existe pas dans le monde réelle, il faut un opérateur intégrateur.

Ensuite, nous avons la *représentation d'état* qui utilise des matrices pour former les équations différentielles comme nous voyons à l'équation 2.7

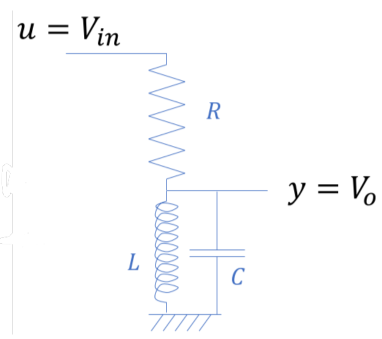


FIGURE 2.1 – Circuit RLC

La dernière représentation type est le *schéma bloc* qui est visuel et qui utilise lui des blocs intégrateurs au lieu de dérivé comme montré à la figure 2.2.

$$\ddot{y} + \frac{1}{CR}\dot{y} + \frac{1}{LC}y = \frac{1}{CR}\dot{u} \tag{2.6}$$

$$\frac{d}{dt}\begin{pmatrix} V_0 \\ I_L \end{pmatrix} = \begin{pmatrix} -1/RC & -1/C \\ 1/L & 0 \end{pmatrix}\begin{pmatrix} V_0 \\ L \end{pmatrix} + \begin{pmatrix} 1/RC \\ 0 \end{pmatrix} u \tag{2.7}$$

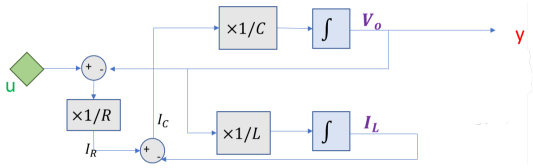


FIGURE 2.2 – Repésentation *bloc* du système à la figure 2.1

### 2.3.3 Équation différentielle entrée-sortie

La forme générale de ces équations est de ce type :

$$\sum_{k=0}^{N} a_k(\frac{d^k}{dt^k})y(t) = \sum_{k=0}^{M} b_k(\frac{d^k}{dt^k})u(t) \tag{2.8}$$

Quelque chose à bien remarquer est que ces équations ne modélise *qu'une partie* d'un système *LIT*. Par exemple, on ne peut pas représenter un *délai* ce qui également rare dans la réalité.

Une chose à remarquer est que l'opérateur *dérivé* peut se "*démultiplier*" et possède une *associativité* et *commutativité*. Ainsi, on peut avoir une représentation dite "*polynomiale*" comme ci-dessus qui est une autre écriture de l'équation 2.8.

$$p(\frac{d}{dt})y(t) = q(\frac{d}{dt})u(t) \tag{2.9}$$

Puis après, pour résoudre ce genre d'équation, on utilise des méthodes classiques vu au cours d'Analyse donc, solution homogène et particulière ...

#### Réponse libre et forcée

Une réponse *libre* est la solution de l'équation homogène, donc quand $u(t) = 0$ à l'équation 2.9 et on garde les **mêmes conditions initiales**. En somme c'est la représentation de l'impact des *CI*.

Une réponse *forcée* est l'équation 2.9 mais avec les *conditions initiales* nulles. Donc on s'intéresse à l'impact de l'entrée sans les conditions initiales. La somme de la réponse *libre* et *forcée* nous donne la réponse générale.

#### Stabilité

On peut avoir une intuition sur la stabilité de notre système en posant $y_H(t)$ qui équivaut à :

$$y_H(t) = \sum_i \alpha_i e^{r_i t} \rightarrow \quad r_i \text{ correspond aux racines de p(z) de 2.9} \tag{2.10}$$

Si la partie réelle de $r_i < 0 \forall i$ alors on a une exponentielle décroissante donc **stable**. On appelle ce genre de système *BIBO* stable ou *Bounded Input Bounded Output*.

En revanche, si $r_i > 0 \exists i$ donc on a au moins une exponentielle croissante créant une *instabilité*.

Si $r_i = 0 \exists i$ on dit qu'on a une *stabilité marginale* ou *instabilité*. Cela dépendera de la **multiplicité** et de $te^{r_i t}$.

#### Linéarité de l'entrée

Avec cette représentation polynomiale, on peut facilement voir qu'on a une linéarité de l'entrée nous permettant de simplifier différent calcul.

#### Avantages et Inconvénients

2 représentations qui sont 2.8 et 2.9. Les avantages :
- Représentation compacte.
- Conditions initiales claires.
- Facile de la transformer dans d'autres représentations.

Les désavantages :
- On peut perdre la représentation physique.

### 2.3.4 Schéma Bloc

Comme son nom l'indique, le schéma bloc utilise des "*blocs*" pour représenter notre système. Ci-dessus on peut voir les composants de base composant ce type de schéma

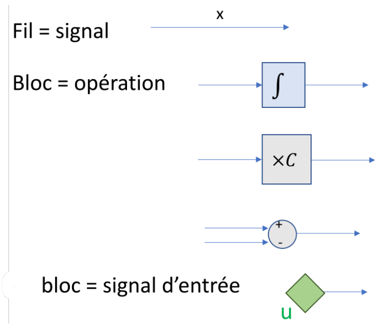


FIGURE 2.3 – Liste reprenant les blocs de base

Dans le cadre des systèmes *LIT*, on se restreint souvent à l'addition, multiplication et intégration. On ne réalise que des opérations *linéaires*.

#### Avantages et Inconvénients

Les avantages :
- Représentation très intuitive.
- Proche de l'implémentation réelle du système.
- On peut plus facilement réfléchir sur le *design* de notre système.
- Très modulaire.

Les désavantages :
- Lien moins clair avec la solution.

De plus, on peut voir comme un *avantage* ou *inconvénient* le fait de pouvoir voir l'évolution des signaux entre chaque bloc plutôt qu'une sorte de boite noire "*entrée-sortie*". De plus, on peut réaliser de bien des manières des circuits.

### 2.3.5 Représentation d'état

Dans ce type de represenation, on a état $x$ qui est un *vecteur* comportant toutes les infos internes de notre système. On entrée $u$ qui est un *signal* extérieur affectant le système. Finalement on a une sortie $y$ qui est un signal qu'on peut accéder depuis l'extérieur.

$$\begin{cases} \text{évolution : } \frac{d}{dt}x(t) = Ax(t) + Bu(t) \\ \text{sortie : } Cx(t) + Du(t) \end{cases} \tag{2.11}$$

**Solution**

La solution *homogène* de $e^{\lambda_i t}$ avec $\lambda_i$ étant les valeurs propres de l'équation. Si la partie réelle des racines est négative pour tout $\lambda_i$.

**État non-unique**

Avec cette représentation, on peut facilement modifier les vecteurs et signaux pour rendre les équations plus simples. Cela n'a aucun impacte et rend les équations plus logiques pour un certain sens "*réel*".

$$\begin{cases} z = Tx \Rightarrow \frac{d}{dt}z = T\frac{d}{dt}x = TAT^{-1}z(t) + TBu(t) \\ y = Cx(t) + Bu(t) = CT^{-1}z(t) + Du(t) \end{cases} \tag{2.12}$$

Si la matrice $A$ est diagonalisable, on peut réaliser un **découplage** et obtenir ainsi un mode dit *découplé*. On peut également faire des *blocs de Jordan* pour diagonaliser le tout :

$$\frac{d}{dt}z_i = \lambda_i z_i + \tilde{B}_i u \tag{2.13}$$

### 2.3.6 Passage de représentation

### 2.3.7 Temps discret

Pour passer du temps continu au temps discret, il faut transformer $\frac{d}{dt}$ en l'opérateur de *décalage* $D$ :

$$Dx[n] = x[n-1] \tag{2.14}$$

Ce qui nous permet d'établir l'équation de *différence*

$$\begin{cases} \sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k u[n-k] \\ p(D)y = q(D)u \end{cases} \tag{2.15}$$

Ce qui nous donne pour solution homogène :

$$y_H[N] = \sum_i c_i r_i^n \tag{2.16}$$

On a une décroissance donc *stabilité* si $|r_i| < 1$ et une croissance si $|r_i| > 1$
De plus, on remplace le bloc *intégrateur* du temps discret en bloc $D^{-1}$. La ré-écriture de l'équation 2.12 en temps discret :

$$\begin{cases} x[n+1] = Ax[n] + Bu[n] \\ y[n] = Cx[n] + Du[n] \end{cases} \tag{2.17}$$

On approxime un système en temps *continu* en temps *discret* ssi :

$$A_d \approx A_c \Delta t \text{ si } \Delta t \text{ petit} \tag{2.18}$$

### 2.3.8 Résumé

1. Réponse impulsionelle, universelle mais peu maniable $\Rightarrow$ On voit que l'entrée et sortie et **Pas de CI**
2. Équation différentielle entrée-sortie $\Rightarrow$ On voit que l'entrée et sortie et on a des **CI**
3. Représentation d'état (matrice) $\Rightarrow$ On voit l'intérieur et on a des **CI**
4. Schéma Bloc (très concret) $\Rightarrow$ On voit l'intérieur et on a des **CI**

### 2.3.9 Existence des systèmes LIT

Une forme usuelle des systèmes LIT dans la vraie vie est de type $\dot{x}(t) = f(x(t), u(t), t)$.
**Invariance temporelle** : tout système fait face à l'usure mais on estime que sur la période d'observation, l'usure est minime et peut être ignorée.
**Linéarité** : aucun système n'est pas linéaire. Cela peut être dû à des *imperfections* ou si on augmente *énormément* l'entrée ce qui change le comportement du système.