MASTER EN SCIENCES MATHÉMATIQUES

Sur la recherche automatique de conditions initiales de systèmes électriques

Tomanos, Dimitri

Award date:
2005

FUNDP
Faculté des Sciences
Département de Mathématique

Rempart de la Vierge, 8
B-5000 Namur Belgique

Sur la recherche automatique de conditions initiales de systèmes électriques

Mémoire présenté pour l'obtention
du grade de
Licencié en Sciences Mathématiques
par

Dimitri TOMANOS

Promoteur : Prof. A. Sartenaer

Année Académique 2004-2005

Je tiens à remercier tout particulièrement Christian Merckx pour sa disponibilité tout au long de cette année ainsi que pour m'avoir permis de réaliser ce travail.

Je tiens également à remercier Annick Sartenaer, promotrice de ce mémoire, pour son aide à la réalisation de ce travail. Je tiens aussi à remercier Jacques Deuze pour son aide très précieuse.

Enfin, je voudrais remercier ma fiancée ainsi que toute ma famille et tous mes amis pour le soutien qu'ils m'ont apporté durant mes années universitaires.

Résumé

Ce travail a été réalisé en collaboration avec Tractebel. Nous proposons plusieurs méthodes afin de trouver automatiquement les conditions initiales de systèmes électriques. Une première méthode arrive à de bons résultats pour de simples systèmes électriques. En faisant plusieurs hypothèses supplémentaires, nous proposons une méthode générale qui donne de bons résultats quel que soit le système.

Abstract

This work was completed in collaboration with Tractebel. We propose several methods in order to automatically find initial conditions of electrical systems. A first method provides good results for simple electrical systems. By making several additional assumptions, we propose a general method which gives good results whatever the system is.

Table des matières

Chapitre 1

Introduction

1.1 Présentation de Tractebel

Tractebel est une des entreprises du groupe SUEZ. Elle s'occupe de l'ingénierie et des services informatiques dans les secteurs :

de l'électricité : production et transport de l'électricité,
du gaz : stockage, transport et traitement du gaz,
de l'infrastructure : infrastructures de transport (rail et eau), barrages, bâtiments et structures complexes.

C'est une entreprise qui compte quelques 3500 collaborateurs et qui est présente dans plus de 20 pays couvrant les 5 continents.

Un des bureaux du groupe Tractebel est Tractebel Engineering qui se trouve à Bruxelles. C'est un des plus importants bureaux de consultance ingéniorale d'Europe. Il propose ses services aux entreprises privées et publiques ainsi qu'aux institutions nationales et internationales.

Un des services de Tractebel Engineering est le "Power System Consulting (PSC)" qui rassemble, sous la direction de Marc Stubbe, une équipe d'une trentaine de personnes. C'est une des seules équipes qui s'occupent du réseau électrique et non des centrales. Ses activités sont regroupées en trois unités :

- La première unité s'occupe principalement des études de systèmes électriques, du système de distribution, d'études de tarification, de la formation, etc.

- Les principales activités de la deuxième unité sont la consultance et la recherche et développement en systèmes électriques.

- La troisième unité s'occupe du développement de logiciels pour l'étude et l'exploitation des réseaux électriques, comme Eurostag par exemple. C'est avec cette troisème unité que ce mémoire a été réalisé afin de rajouter une fonctionnalité au logiciel Eurostag.

1.2 Présentation du logiciel Eurostag

Les réseaux électriques deviennent de plus en plus complexes. Les décisions à prendre dépendent de plusieurs facteurs comme les contraintes économiques, environnementales, etc. Afin de diminuer les investissements, un programme de simulation est nécessaire pour produire rapidement des modèles efficaces. Tractebel et Electricité de France ont donc, en 1988, conjointement développé le logiciel Eurostag.

Eurostag permet, par exemple, de visualiser le comportement du réseau jusqu'à ce qu'il retourne à un état d'équilibre et ce, quelle que soit la taille du système ou la perturbation. Eurostag est également très agréable à utiliser car il permet de modéliser les systèmes de manière graphique et on diminue donc le risque d'erreur de programmation car les erreurs sont beaucoup plus visibles. De plus, afin de faciliter l'interprétation, il fournit une batterie d'outils graphiques permettant une meilleure analyse des résultats.

Eurostag est séparé en plusieurs modules :

- Un module qui permet de convertir des données fournies dans un format international dans le format utilisé par le logiciel.

- Un éditeur graphique et un éditeur de fichiers permettant d'introduire les données du système électrique.

- Un éditeur de modèles graphiques permettant la représentation des régulations sous un format graphique appelé *macrobloc* et qui permet également la création de nouveaux modèles. Une fois le macrobloc créé, on doit le compiler et une autre partie du module, le programme d'analyse topologique, vérifie que tout est bien construit et crée ensuite les fichiers utilisés par les modules suivants.

- Le coeur du logiciel qui contient les programmes de calcul de load flow, c'est-à-dire un calcul de ce qui se passe sur le réseau en supposant une production et une consommation constante (ce qui se fait en résolvant les équations de Kirchoff associées au réseau) et un programme de simulation qui montre l'évolution de certaines variables du réseau lorsqu'un événement survient. On peut alors simuler, par exemple, des

pannes et voir comment évolue le réseau.

- Un post-processeur graphique qui permet d'analyser les résultats en fournissant des figures avec les courbes d'évolution des variables.

- Un module permettant de créer des tableaux de résultats que l'on peut filtrer, classer, etc.

En conclusion, lorsque nous avons un système électrique à étudier, nous le construisons avec les différents éditeurs. Nous utilisons ensuite le quatrième module afin de faire la simulation. Nous présentons alors les résultats par le cinquième module et le dernier module permet de créer des tableaux à insérer dans des rapports.

1.3 Présentation du sujet

Les équations d'un système électrique se décomposent en deux parties :

- Les équations du réseau qui sont des équations algébriques. Elles correspondent aux équations de Kirchoff.

- Les équations des centrales qui sont différentielles et algébriques. Celles-ci se décomposent en trois parties :

  - Les équations de Park qui sont différentielles et algébriques et qui ont pour but de modéliser la machine.
  - Les équations d'interconnexion qui sont algébriques et qui permettent de lier la machine au réseau.
  - Les équations des régulations qui permettent de réguler la machine, par exemple, de contrôler sa vitesse.

Soit $X = (x_i)_{i\in(1,\ldots,n)}$ le vecteur contenant l'ensemble des variables présentes dans le système électrique. Si, dans celui-ci, il y a p équations algébriques et q équations différentielles, alors le système correspondra au système d'équations :

$$\begin{cases} f_k(x) = 0 & k = 1,\ldots,p \\ g_j(x) + h_j(\dot{x}) = 0 & j = 1,\ldots,q, \end{cases} \tag{1.1}$$

où $\dot{x}$ est la dérivée temporelle d'ordre un de x.

Nous allons faire l'hypothèse que les systèmes électriques, au temps initial, sont en situation stationnaire. D'un point de vue physique, cela veut dire qu'on se trouve dans une situation d'équilibre. On suppose de plus que cette situation dure depuis suffisamment longtemps. D'un point de vue mathématique, cela se traduit par le fait que toutes les dérivées temporelles sont nulles. Cette hypothèse est sensée. En effet, il est raisonnable de penser qu'au temps initial, le système est en équilibre et qu'il n'y a pas de perturbations. Le système devient alors :

$$\begin{cases} f_k(x) & = & 0 & \quad k & = & 1,\ldots,p \\ g_j(x)+h_j(0) & = & 0 & \quad j & = & 1,\ldots,q. \end{cases} \tag{1.2}$$

Nous obtenons donc un système d'équations algébriques à résoudre. Nous savons que la résolution du système (1.1) nécessite la connaissance de q valeurs initiales. En effet, pour résoudre une équation différentielle d'ordre un, il est nécessaire de connaître une valeur initiale et comme il y a q équations différentielles dans le système, on doit connaître q valeurs initiales. La résolution du système (1.1) permet de trouver ces valeurs initiales.

Actuellement, pour les trouver, les ingénieurs de Tractebel écrivent les équations du système, remplacent les dérivées par zéro et ensuite résolvent ce système à la main. Le but de ce mémoire est, à partir d'un système électrique quelconque, de trouver *automatiquement* l'ensemble des valeurs initiales.

Chapitre 2

Méthode pour blocs simples

2.1 Macroblocs et blocs élémentaires

Un *macrobloc* est une réprésentation graphique des équations d'une égulation. Il représente la dynamique de la machine. Les concepteurs d'Eurostag utilisent cette notion pour représenter les régulations des systèmes électriques dans le logiciel Eurostag.

2.1.1 Blocs élémentaires

Pour représenter graphiquement un macrobloc, on utilise la notion de *blocs élémentaires*. Ceux-ci correspondent à des fonctions communément utilisées en automatique. Ils ont certaines caractéristiques communes :

- Il y a entre 0 et n entrées.
- On applique à ces entrées une certaine fonction, laquelle a besoin parfois de certains paramètres.
- On obtient une sortie.

Un bloc élémentaire est une représentation graphique de l'équation $y = f(x)$ où x est le vecteur contenant l'ensemble des entrées du bloc, y la sortie et f la fonction représentée par le bloc élémentaire. Voici quelques exemples de blocs élémentaires.

La figure 2.1 ci-dessous montre un exemple de bloc élémentaire, *le bloc Constant*. Ce bloc n'a pas d'entrée et sa sortie est toujours la valeur constante K. La fonction mathématique correspondante est donc $y = K$.

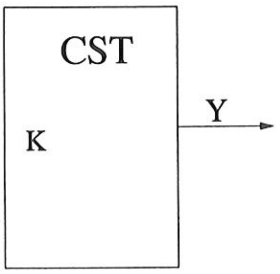

FIG. 2.1 – Le bloc constant

La figure 2.2 est le *bloc Gain*. La fonction d'un bloc Gain est $y = Kx$. C'est en fait un bloc permettant de multiplier une variable par un scalaire donné. On voit que ce bloc a une entrée et est la représentation d'une fonction algébrique.

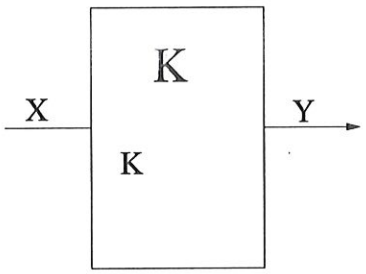

FIG. 2.2 – Le bloc constant

La figure 2.3 est le bloc de fonction $y = \cos(x)$ et sa sortie sera donc l'évaluation par la fonction cosinus de l'entrée. Le bloc possède une entrée et est la représentation d'une fonction trigonométrique.

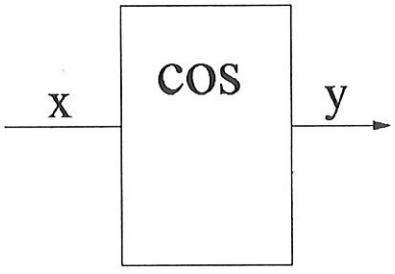

FIG. 2.3 – Le bloc cosinus

La figure 2.4 est le bloc de fonction $y = \sum_i a_i x_i + b$ et sa sortie sera donc la somme pondérée des entrées plus éventuellement un scalaire b. Le bloc possède plusieurs entrées.

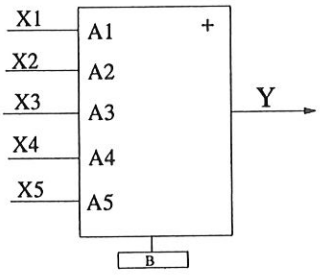

FIG. 2.4 – Le bloc sommateur

La figure 2.5 est le bloc de fonction $y = \frac{A1X1}{A2X2} + b$ et sa sortie sera donc le quotient des deux entrées plus éventuellement un scalaire b. Le bloc possède deux entrées. Les lettres N et D présentes à côté des entrées signalent quelle entrée est le numérateur et quelle entrée est le dénominateur.

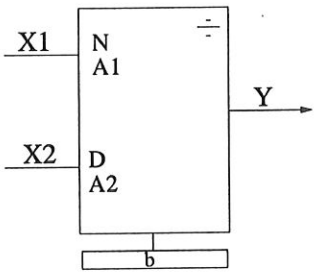

FIG. 2.5 – Le bloc diviseur

Il existe de nombreux blocs (une cinquantaine) mais il serait fastidieux de tous les représenter ici. Ils peuvent tous être trouvés dans [1].

Les blocs présentés ci-dessus sont la représentation d'une fonction algébrique ou trigonométrique, mais on peut aussi représenter certaines équations différentielles qui apparaissent très souvent dans les régulations et on obtient ainsi une catégorie très importante de blocs élémentaires, *les blocs différentiels*. Ils peuvent être reconnus par le fait qu'ils ont un petit carré noir en haut à droite du bloc. La figure 2.6 ci-dessous est un exemple de bloc différentiel.

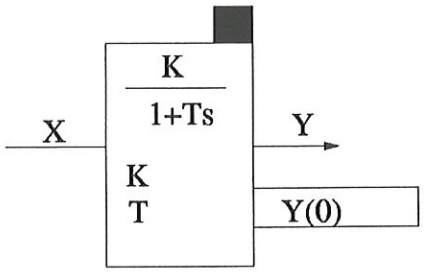

FIG. 2.6 – Le bloc simple lag

La figure 2.6 est le bloc *simple lag*. Il correspond à l'équation différentielle

$$\dot{y} = \frac{K}{T}x - \frac{1}{T}y. \tag{2.1}$$

L'équation présente dans le bloc est la transformée de Laplace de l'équation (2.1). Il y a bien évidemment la contrainte $T \neq 0$. Les paramètres K et T sont des paramètres à fixer par l'utilisateur et $Y(0)$ est la valeur initiale *à trouver*. De manière générale, les valeurs initiales seront dans un petit rectangle en bas à droite du bloc. Il est possible de construire, dans le model editor, un bloc dont la valeur initiale n'est pas dans ce petit rectangle mais nous exclurons ce cas particulier car on peut facilement se ramener à un bloc avec la valeur initiale dans le rectangle en bas à droite. Tous les blocs différentiels ont tous une valeur initiale à déterminer.

La figure 2.7 est la représentation du bloc *Lead Lag*. La fonction du bloc est :

$$\dot{y} = -\frac{y}{T2} + \frac{KT1}{T2}\dot{x} + \frac{K}{T2}x. \tag{2.2}$$

L'équation présente dans le bloc est la transformée de Laplace de l'équation différentielle (2.2). Nous avons la contrainte $T2 \neq 0$. Les paramètres K, $T1$ et $T2$ sont aussi à fixer par l'utilisateur. La valeur $Y(0)$ présente dans le petit rectangle en-bas à droite est la valeur initiale du bloc qui doit être trouvée.

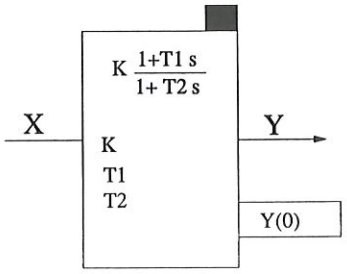

FIG. 2.7 – Le bloc Lead Lag

La figure 2.8 est le bloc de fonction

$$\ddot{y} = \frac{1}{B2}(-B1\dot{y} - y + A2\ddot{x} + A1\dot{x} + A0x) \tag{2.3}$$

L'équation présente dans le bloc est la transformée de Laplace de l'équation différentielle (2.3). Nous avons la contrainte $B2 \neq 0$. Les paramètres $A0$, $A1$, $A2$, $B1$ et $B2$ sont aussi à fixer par l'utilisateur. La valeur $Y(0)$ présente dans le petit rectangle en-bas à droite est la valeur initiale du bloc qui doit être trouvée. Ce bloc a la caractéristique intéressante d'être le seul bloc qui a des dérivées temporelles d'ordre plus grand que un. Les petits rectangles noirs entrecoupés du rectangle blanc qui se trouvent en haut à droite du bloc signifient que le bloc représente une équation d'ordre deux.

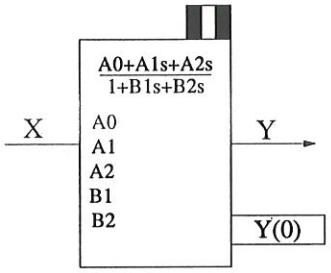

FIG. 2.8 – Le bloc Second Order

La composée de plusieurs blocs élémentaires permet d'obtenir toute sorte de fonctions mathématiques, linéaires ou non. Par exemple, la composée d'un bloc cosinus et d'un bloc exponentiel représentera la fonction $y = \exp(\cos(x))$. Un macrobloc est une composée de plusieurs blocs (parfois même une centaine voire plus).

2.1.2 Variables d'état

Les variables d'état sont des variables qui permettent de définir l'état du système. Pour chaque variable d'état, on a une équation. On aura donc une série d'équations définissant le système. C'est ce qu'on appelle le *schéma principal.* Il y a plusieurs types de variables d'état :

- Les premières variables d'états sont les *variables qui viennent des équations du réseau.* Elles viennent donc des équations de Kirchoff et réprésentent la tension aux différents noeuds du réseau.

- Ensuite, il y a les *variables d'interface prédéfinies.* Elles correspondent aux équations de Park et aux équations d'interconnexion. Ce sont en général des variables propres à la machine et qui se retrouvent dans un grand nombre de machines

comme, par exemple, le couple moteur.

- Il y a ensuite les *variables différentielles des régulations* qui correspondent aux sorties des blocs différentiels dans les macroblocs des régulations.

- Enfin, il y a les *variables d'interface utilisateur*. L'utilisateur peut définir la sortie de n'importe quel bloc comme étant une variable d'interface. Pour chaque variable d'interface, on aura une équation supplémentaire au système. L'existence de cette équation supplémentaire n'a aucune explication mathématique ou physique. On désire uniquement rendre cette variable accessible dans d'autres macroblocs ou donner accès aux modules suivants du logiciel Eurostag à cette variable.

Les deux premiers types de variables sont déterminés par deux autres méthodes. La première méthode est ce qu'on appelle le load flow et elle permet de connaître le premier type de variables. Le deuxième méthode est l'initialisation de la machine qui permet de déterminer les variables d'interface prédéfinies. Ces variables étant connues, on pourra les utiliser afin de résoudre les équations de la régulation.

2.2 Séparation en équations

Le macrobloc représente un système d'équations algébro-différentiel (contenant des équations algébriques et des équations différentielles). A partir d'un macrobloc donné, il est possible de retrouver toutes les équations liées à la régulation de la machine. En effet, nous savons que chaque variable d'état va définir une équation. Nous allons donc "couper" le macrobloc à chaque fois que nous allons rencontrer une variable d'état. De cette façon, chaque "morceau" sera une équation qui sera soit algébrique, soit différentielle, suivant que le dernier bloc est différentiel ou non.

La séparation en équations sera utilisée pour la méthode de résolution de macroblocs contenant uniquement des blocs simples. On pourra couper le macrobloc et donc avoir une nouvelle équation dès que l'on rencontre une variable d'état. Nous pouvons remarquer que, dans la représentation graphique utilisée, l'utilisateur signalera que la sortie d'un bloc est une variable d'interface utlisateur ou prédéfinie en notant, au dessus de la flèche de sortie du bloc, un nom correspondant au nom de la variable d'interface.

Pour illustrer cette notion, nous avons représenté, à la figure 2.9, un macrobloc très simple qui n'a aucune signification électrique mais qui permettra de comprendre plus aisément cette notion.

Dans ce macrobloc, il n'y a que trois variables d'état , z, x et y. Nous allons parcourir le macrobloc et couper après chaque variable d'état. On obtiendra alors le schéma de la

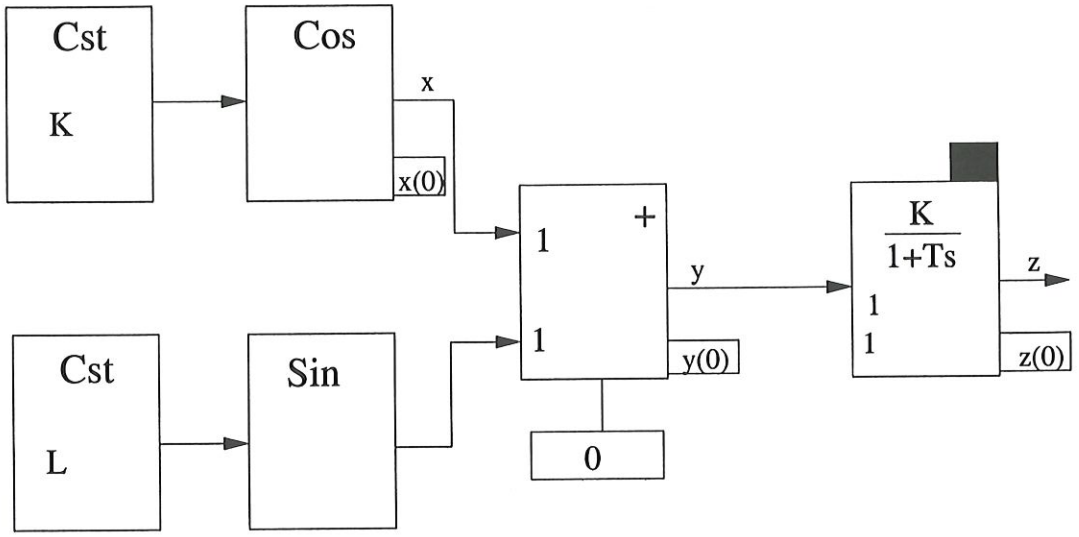

FIG. 2.9 – Illustration d'un macrobloc simple

figure 2.10. Les gros traits correspondent à une cassure du macrobloc.

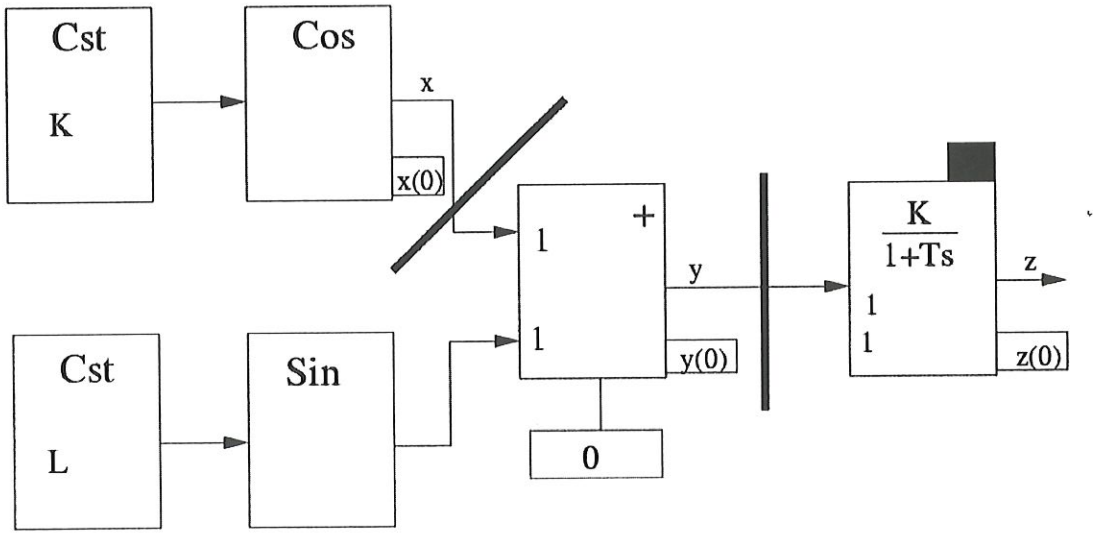

FIG. 2.10 – Illustration du macrobloc coupé

Nous pouvons constater que nous avons alors bien trois équations distinctes qui sont :

$$\begin{aligned} x &= cos(K) \\ y &= x + sin(L) \\ \dot{z} &= y - z. \end{aligned}$$

2.3 Recherche de valeurs initiales

Comme cela a déjà été dit prédemment, les blocs différentiels ont une valeur initiale qui doit être déterminée. En effet, ils correspondent à une équation différentielle et pour la résoudre, il est nécessaire d'avoir une condition initiale. Cependant, certains blocs ont aussi une valeur initiale qui doit être trouvée. C'est le cas, par exemple, du bloc *Set Point*. Ce bloc a la caractéristique intéressante de fournir une valeur initiale sans apporter d'équations. La sortie du bloc n'est donc ni une variable d'interface, ni une variable d'état, mais juste une variable dont il faut trouver la valeur initiale. Sa représentation peut être trouvée à la figure 2.11.

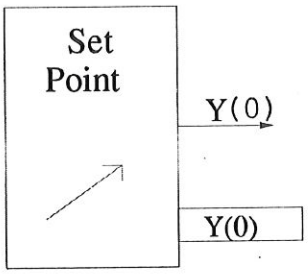

FIG. 2.11 – Le bloc Set Point

Pour résoudre le système, il est nécessaire de connaître les valeurs initiales des variables d'état et d'interface. Actuellement, les ingénieurs utilisant le logiciel Eurostag doivent construire le *schéma initial* qui permet de calculer ces valeurs initiales. Ils doivent analyser chaque régulation séparément et trouver ces valeurs en utilisant les données de la machine.

2.4 Remplacement des blocs

Comme nous avons fait l'hypothèse d'une situation stationnaire, les équations des blocs différentiels se simplifient fortement. En effet, comme on remplace les dérivées par zéro, les équations différentielles vont devenir de simples équations algébriques. Illustrons ce principe sur plusieurs blocs élementaires.

- Reprenons tout d'abord le bloc simple lag dont la représentation se trouve à la figure 2.6 et dont l'équation se trouve en (2.1). Au temps initial, la dérivée s'annule et on a donc :

$$\begin{aligned} 0 &= \tfrac{K}{T}x(0) - \tfrac{1}{T}y(0) \\ \Leftrightarrow \tfrac{1}{T}y(0) &= \tfrac{K}{T}x(0) \\ \Leftrightarrow y(0) &= Kx(0). \end{aligned}$$

On pourra donc remplacer le bloc simple lag par un bloc Gain de paramètre K au temps initial car l'équation du bloc simple lag au temps initial est la même que l'équation du bloc multiplicateur.

- Reprenons ensuite le bloc lead lag dont la représentation se trouve à la figure 2.7 et dont l'équation se trouve en (2.2). Au temps initial, les dérivées s'annulent et nous obtenons alors :

$$\begin{aligned} 0 &= -\tfrac{y(0)}{T2} + \tfrac{K}{T2}x(0) \\ \Leftrightarrow \tfrac{y(0)}{T2} &= \tfrac{K}{T2}x(0) \\ \Leftrightarrow y(0) &= Kx(0). \end{aligned}$$

On pourra donc remplacer le bloc lead lag par un bloc Gain de paramètre K au temps initial car l'équation du bloc lead lag au temps initial est la même que l'équation du bloc multiplicateur.

- Reprenons enfin le bloc second order dont la représentation se trouve à la figure 2.8 et dont l'équation se trouve en (2.3). Au temps initial, les dérivées s'annulent et nous obtenons alors :

$$\begin{aligned} 0 &= \tfrac{1}{B2}(-y(0) + A0x(0)) \\ \Leftrightarrow \tfrac{y(0)}{B2} &= \tfrac{A0}{B2}x(0) \\ \Leftrightarrow y(0) &= A0x(0). \end{aligned}$$

On pourra donc remplacer le bloc second order par un bloc Gain de paramètre A0 au temps initial car l'équation du bloc second order au temps initial est la même que l'équation du bloc multiplicateur.

On voit donc que l'effet de cette situation stationnaire dans le macrobloc sera le remplacement des blocs différentiels par des blocs de multiplication par une constante. On peut généraliser ce principe pour remplacer quelques autres blocs par des blocs beaucoup

plus simples. Dans l'exemple exposé à la figure 2.9, on peut remplacer le bloc simple lag par un bloc multiplicateur et on obtient alors le schéma de la figure 2.12

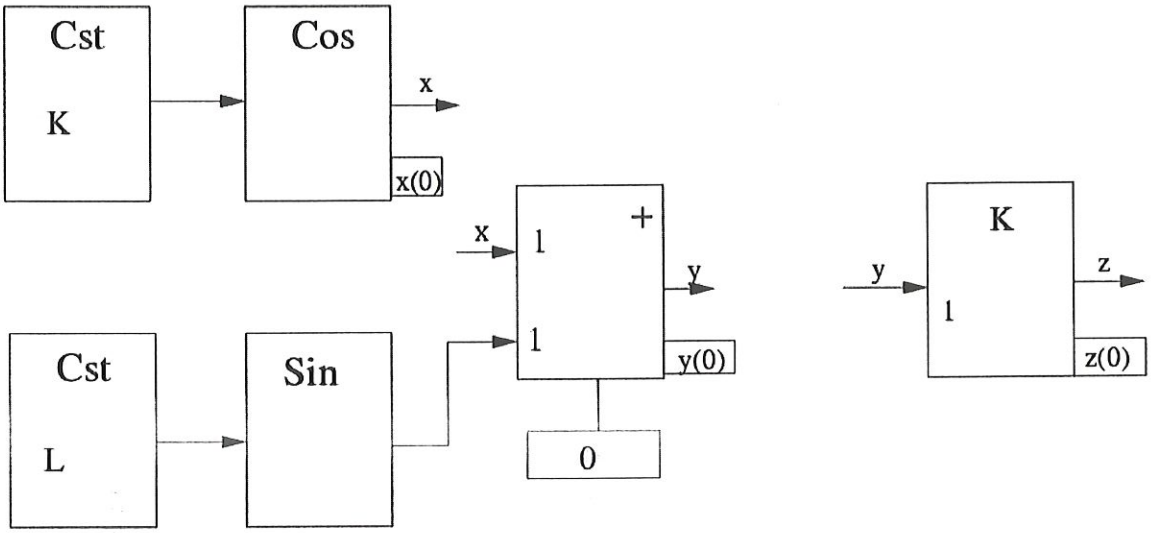

FIG. 2.12 – Remplacement des blocs dans un macrobloc simple

Les équations représentées par ce schéma sont le système suivant :

$$\begin{aligned} x &= cos(K) \\ y &= x + sin(L) \\ y &= z, \end{aligned}$$

ce qui correspond bien aux équations du schéma initial.

2.5 Notions de montée et de descente

Une fois que les blocs ont été remplacés, il faut résoudre les équations afin de trouver les valeurs initiales. Pour cela, on va utiliser deux notions :

- Supposons que l'on ait une équation et que l'on connaisse l'entrée du premier bloc de l'équation ou que ce bloc donne une valeur connue. La sortie du dernier bloc est à déterminer, parce que c'est une variable d'interface ou une variable d'état dont la valeur initiale n'a pas encore été déterminée. On part alors de l'entrée du premier bloc faisant partie du groupe de blocs représentant l'équation, on évalue chaque bloc rencontré pour finalement arriver à la sortie du dernier et ainsi trouver la

valeur initiale inconnue. Nous dirons alors que *nous descendons les blocs.*

Supposons, par exemple, que l'on ait une équation d'un macrobloc, que celle-ci est représentée par la suite de blocs présenté à la figure 2.13 et que l'on désire connaître la valeur de la sortie.

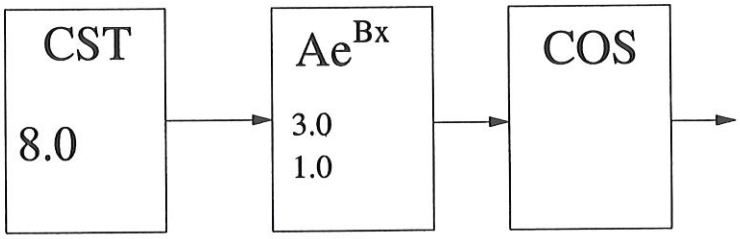

FIG. 2.13 – Illustration d'une équation

On part du bloc CST qui nous donne une constante, on évalue le bloc suivant, la sortie du deuxième bloc est alors $3e^8$ et ensuite on évalue le troisième bloc, la sortie vaut donc $cos(3e^8)$. Ce raisonnement est représenté à la figure 2.14.

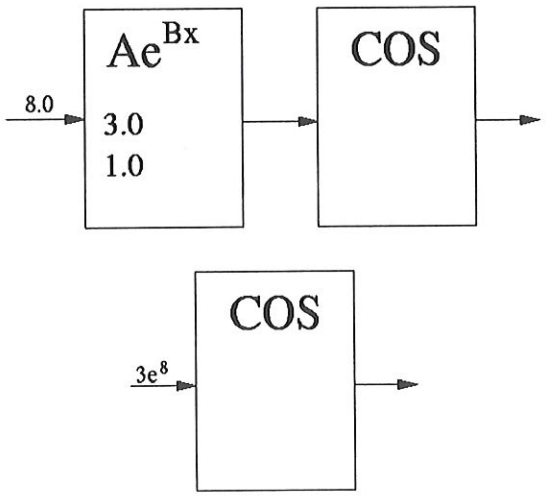

FIG. 2.14 – Illustration du fonctionnement de la descente

- Supposons maintenant que l'on ait une équation, que l'on connaîsse déjà la sortie du dernier bloc (parce que c'est une variable prédéfinie), et que l'on désire connaître l'entrée de cette équation. On remonte alors, si possible, les blocs un à un afin de déterminer la valeur initiale inconnue. Pour pouvoir remonter un bloc, il faudra prendre la fonction inverse du bloc et ce sera parfois impossible. Si c'est possible,

nous dirons que *nous remontons les blocs.*

Supposons, par exemple, que l'on ait une équation d'un macrobloc, que celle-ci est représentée par la suite de blocs présentée à la figure 2.15 et que l'on désire connaître la valeur V1.

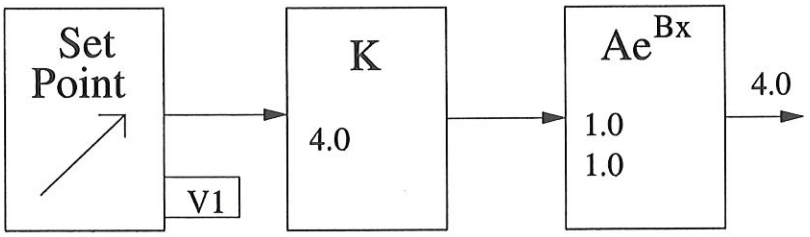

FIG. 2.15 – Illustration d'une équation

Supposons que l'on connaisse la valeur de la sortie, 4.0. On remonte alors le dernier bloc puisque la fonction exponentielle admet un inverse. On a donc que l'entrée du bloc vaut $ln(4.0)$. On remonte ensuite le deuxième bloc puisque la fonction de multiplication par une constante admet un inverse et on a donc que l'entrée du deuxième bloc vaut $\frac{1}{4}ln(4.0)$. Or cette entrée est aussi la valeur inconnue qu'on a donc déterminée. Ce raisonnement est repésenté à la figure 2.16.

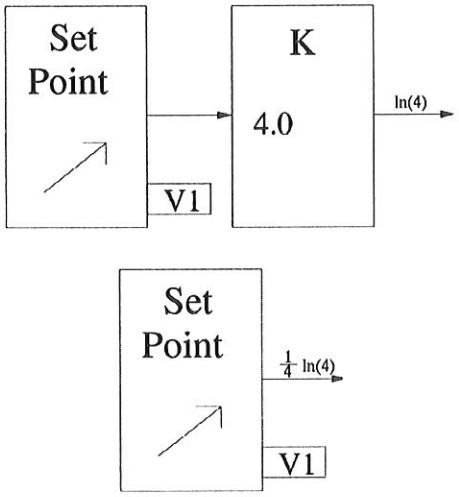

FIG. 2.16 – Illustration du fonctionnement de la remontée

Quand nous essayerons de trouver l'ensemble des valeurs initiales d'un macrobloc, nous utiliserons ces notions simultanément. En effet, on partira des entrées du macrobloc

et de ses sorties et nous descendrons et remonterons les blocs jusqu'à ce que chaque valeur initiale ait été trouvée. On peut s'imaginer que cela n'est pas toujours possible. En effet, on peut arriver à des équations qui seront couplées et qu'on ne pourra pas résoudre séparément. Il faudra alors résoudre un système d'équations. Cela sera expliqué au chapitre 4.

Si nous utilisons une illustration matricielle du système, alors la descente correspondra à une substitution progressive et la remontée à une substitution régressive. Pour que la descente soit possible, il faut que le bloc supérieur de la matrice soit triangulaire inférieur. Pour que la remontée soit possible, il faut que le bloc inférieur de la matrice soit triangulaire supérieur. Si on a un bloc dans la matrice qui est quelconque, cela correspondra à un système d'équations couplées. Voici une représentation matricielle d'un système. La descente est possible pour trois équations et la remontée pour deux équations. Ensuite, il faudra résoudre un système. Les x représenteront les éléments non nuls de la matrice.

$$\begin{pmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & x & x & x & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x \end{pmatrix}$$

2.6 Classification des blocs élementaires

Nous allons classifier les blocs élémentaires afin de pouvoir dégager certains points communs, ce qui nous sera très utile pour la création du schéma initial.

- D'une part, comme dit précédemment, l'utilisateur peut définir la sortie de n'importe quel bloc comme étant une variable d'interface. Ces blocs donnent alors une équation et sont donc à traiter différemment des autres. Cependant, on ne peut classer ces blocs dans une catégorie à part puisque chaque bloc peut faire partie de cette catégorie. Néanmoins, il faut garder à l'esprit que dès que la sortie d'un bloc sera définie de la sorte, alors ce bloc sera à traiter de manière différente.

- Tout d'abord, il est intéressant de regarder quels blocs demandent le moins de travail. On peut alors regrouper toute une série de blocs que l'on appellera *blocs à valeur connue*. On mettra dans cette catégorie le bloc *Constant* et tous *les blocs de mesure* car ils ne font que donner une valeur connue et n'apporte pas d'équation. Par blocs de mesure, nous entendons les blocs n'ayant aucune entrée et qui donnent la mesure d'une certaine variable de la machine. Pour ces blocs, il n'y a rien à calculer, rien à déterminer.

- D'autre part, on peut séparer, tout d'abord, les blocs restants en trois grandes catégories (hormis certaines exceptions qu'on devra traiter individuellement) : les blocs différentiels, les blocs dont la fonction correspondante est une fonction algébrique (ou trigonométrique) et les blocs dont la fonction est une fonction logique.

  1. Premièrement, en ce qui concerne les blocs différentiels, on peut les séparer en deux types :

     - Les blocs différentiels *non limités* : ce sont les blocs dont l'équation reste la même quelle que soit l'entrée.

     - Les blocs différentiels *limités* : la sortie est limitée à un intervalle donné. Quelle que soit la valeur de l'entrée, la valeur de la sortie doit appartenir à cet intervalle. Ces blocs seront traités au chapitre suivant.

  2. Deuxièmement, les blocs algébriques (dont la fonction est une fonction algébrique) peuvent être séparés en trois catégories :

     - Les blocs algébriques *simples dont la fonction admet un inverse.* C'est-à-dire les blocs dont la fonction est inversible comme la somme, l'exponentielle.

     - Les blocs algébriques *simples sans inverse.* C'est-à-dire des blocs dont la fonction n'est pas inversible comme la valeur absolue, l'arccosinus, etc. Ils seront traités au chapitre 3. On voit bien qu'il est nécéssaire de faire la distinction entre les blocs dont la fonction admet un inverse et ceux dont la fonction n'en admet pas puisqu'ils seront à traiter différemment si une remontée à travers ces blocs s'avérait nécéssaire.

     - Les blocs algébriques *avec alternatives* : selon l'entrée, on aura plusieurs équations possibles. Ils seront traités également au chapitre 3.

     Nous allons regrouper les blocs algébriques avec alternatives et les blocs algébriques simples sans inverse en une seule catégorie car ces deux types de blocs sont à traiter de la même manière (voir chapitre 3). Ces blocs apportent une équation mais pas de valeur initiale à déterminer.

  3. Troisièmement, les blocs logiques seront les blocs dont la fonction est une fonction logique. C'est-à-dire que, quelle que soit l'entrée, la sortie est toujours soit 0 soit 1. Ils apportent une équation mais pas de valeur initiale à déterminer (sauf exception).

- Il ne reste alors plus que trois blocs à traiter :

  - Le bloc Set point doit être classé séparément. En effet il est le seul à avoir une valeur initiale à déterminer et à ne pas apporter d'équation.

  - Les deux autres blocs sont *Hysteresis* et *Switching with Hysteresis*. On les classera ensemble dans la catégorie *blocs difficiles* car ils sont difficiles à traiter et n'ont aucun paramètre à déterminer.

Une première partie de ce mémoire a été l'élaboration de cette classification. Chaque bloc a dû être analysé et comparé afin de cerner les points communs les plus importants pour déduire un classement. Ce tri a ensuite été proposé et approuvé par Tractebel. Il va de soi que cette analyse fastidieuse n'a pas lieu d'être dans cette section. Cependant, les critères principaux de tri ont été exposés dans l'explication des différentes catégories. L'ensemble des blocs énoncés ci-dessous peuvent se trouver dans [1].

1. Les blocs à valeur connue

   - Active power
   - Constant
   - Field current
   - Reactive power
   - Reference frequency
   - Terminal voltage

2. Le Set point

3. Les blocs algébriques simples avec inverse

   - Continuous input delay
   - Divider
   - Exponential
   - Function
   - Gain
   - Inverse Function
   - Multiplier
   - Pulse
   - Relay
   - Relay with delay
   - Summer

4. Les blocs différentiels non limités

   - Random input delay
   - Derivative lag
   - Integrator
   - Integrator follower
   - Lead lag filter
   - Second order
   - Simple lag

5. Les blocs algébriques simples sans inverse et les blocs algébriques avec alternatives.

   - Abs
   - Arccos
   - Arcsin
   - Arctan
   - Cos
   - Deadband
   - Limiter
   - Max
   - Min
   - Power
   - Sin
   - Square root
   - Tan
   - Variable limiter
   - Zero set counter

6. Les blocs différentiels limités

   - Limited integrator
   - Limited lead lag filter
   - Limited simple lag

7. Les blocs logiques

   - Logical and
   - Monostable
   - Or
   - Or with hold
   - Reset-set
   - Set-reset

8. Les blocs difficiles

   - Hysteresis
   - Switching with hysteresis

2.7 Analyse des blocs simples

Nous allons à présent analyser individuellement chaque bloc afin de voir leurs caractéristiques particulières et ainsi dégager une méthode de résolution. Nous allons commencer par traiter les blocs *simples*, c'est-à-dire les blocs dont l'équation ne change pas selon l'entrée. Les blocs à valeur connue, les blocs algébriques simples, le bloc *Set Point* et les blocs différentiels non limités font partie de cette catégorie et ce sont les seuls. Les autres blocs seront traités au chapitre suivant. Par souci de clarté, nous allons traiter ici uniquement deux blocs représentatifs mais l'analyse complète de chaque bloc peut être trouvée en annexe. Notons que le bloc Arctan, les blocs logiques et les blocs difficiles n'ont pas été analysés mais ils n'apportent pas grand chose de plus. L'important était d'avoir suffisamment de blocs pour trouver une méthode de résolution et pas l'analyse des blocs. Tous les blocs seront traiter de la même façon, à savoir :

Description Certains blocs ont une équation différente quand on est au temps initial. C'est le cas des blocs différentiels par exemple dont l'équation change à cause de la situation stationnaire. On décrira dans cette partie de l'analyse l'équation modifiée du bloc ainsi que quelques hypothèses qui sont faites.

Valeur Initiale ? Certains blocs ont une valeur initiale à déterminer. C'est le cas par exemple des blocs différentiels. On écrira dans cette partie la valeur initiale à déterminer.

Remplacement L'équation de certains blocs change en faisant l'hypothèse d'une situation stationnaire. On écrira ici les blocs qui remplaceront le bloc de départ pour obtenir l'équation du bloc au temps initial.

Sens On décrira dans cette partie les sens de parcours possibles du bloc. Comme on l'a vu, on peut soit remonter, soit redescendre une équation. Mais, parfois, on ne peut pas remonter certains blocs. Nous l'écrirons alors dans cette partie de l'analyse.

Résolution On écrira ici les calculs effectués pour obtenir l'équation au temps initial à partir de l'équation générale du bloc ainsi que les calculs pour savoir comment on peut trouver une entrée à partir des autres entrées et de la sortie du bloc.

Nous allons commencer par analyser le bloc Gain. Sa représentation peut être trouvée à la figure 2.2 et, pour rappel, il correspond à l'équation $y = Kx$. Nous obtenons l'analyse suivante :

Description On a $y(0) = Kx(0)$. Le cas ou $K = 0$ est caduque car cela supposerait que la sortie est nulle quelle que soit l'entrée, ce qui n'a pas de grande utilité. Nous pouvons donc supposer que $K \neq 0$.

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution On a $y(0) = Kx(0)$ dans le cas où on descend le bloc. Par contre, si on le remonte, on a $x(0) = \frac{1}{K}y(0)$.

Nous allons maintenant analyser un bloc différentiel. Nous avons choisi d'analyser le bloc simple lag car c'est un bloc fort récurrent dans les régulations analysées. La représentation graphique peut être trouvée à la figure 2.6, son équation peut être trouvée en (2.1). L'explication de l'équation au temps initial peut être trouvée dans la section 2.3. Voici l'analyse déduite :

Description On a $y(0) = Kx(0)$.

Valeur Initiale ? Oui, $y(0)$.

Remplacement Ce bloc doit être remplacé par un bloc Gain avec un paramètre K.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution Simple.

2.8 Méthode déduite pour des blocs simples

De par l'analyse des blocs et par les notions expliquées précédemment, nous avons pu déduire une méthode de résolution pour les macroblocs uniquement constitués de blocs simples :

- On sépare le macrobloc en équations.

- En remplacant les blocs élémentaires par leur équivalent au temps initial (principalement les blocs différentiels), on peut écrire les équations du schéma d'initialisation. Tous les blocs de remplacement peuvent être trouvés dans l'analyse des différents blocs en annexe.

- Les blocs à valeur connue apportent des valeurs connues pour descendre les équations dans lesquelles ils interviennent.

- Par ailleurs, les variables d'interface prédéfinies ont une valeur initiale connue (de par l'initialisation de la machine). On pourra donc les remplacer dans les équations. Ces variables d'interface peuvent apparaître en fin d'équation. Elles permettent alors de remonter les équations.

- On peut maintenant résoudre les équations. On cherche les équations ne faisant intervenir qu'une seule valeur initiale. On trouve cette valeur initiale en descendant ou en remontant l'équation, selon le cas. Ensuite, on remplace les valeurs initiales trouvées dans les autres équations. On fait cela jusqu'à ce que toutes les valeurs initiales soient trouvées.

2.9 Application de la méthode à la régulation gover3

On va appliquer cette méthode à la main à une régulation simple, gover3.frm. Cette régulation peut être trouvée dans [2]. Le macrobloc de la régulation se trouve à la figure 2.17.

Il y a dans cette régulation un bloc qui n'a pas encore été traité, le bloc 5 . Néanmoins, vu la difficulté de trouver une régulation simple, faisons comme si ce bloc n'intervenait pas. Ce bloc sera traité au chapitre suivant. Tous les blocs présents dans le macrobloc sont expliqués dans la première section de ce chapitre. Notons que les variables d'interface CM et OMEGA sont des variables d'interface prédéfinies, on connaît donc la valeur initiale de ces variables.

L'objectif de la méthode est de déduire le schéma initial, c'est-à-dire de trouver le système algébrique permettant de déduire les valeurs initiales et de le résoudre. Nous avons sept valeurs initiales à trouver, les valeurs initiales des blocs 2, 3, 4, 6, 7, 8 et 9.

Nous allons séparer tout d'abord le macrobloc en équations. Nous partons de l'entrée du macrobloc, $OMEGA$, nous arrivons au bloc numéro 2 et nous obtenons donc une équation car la sortie est une variable d'état. Nous arrivons ensuite au bloc 4 et nous avons, à nouveau, une nouvelle équation car la sortie de ce bloc est définie comme étant

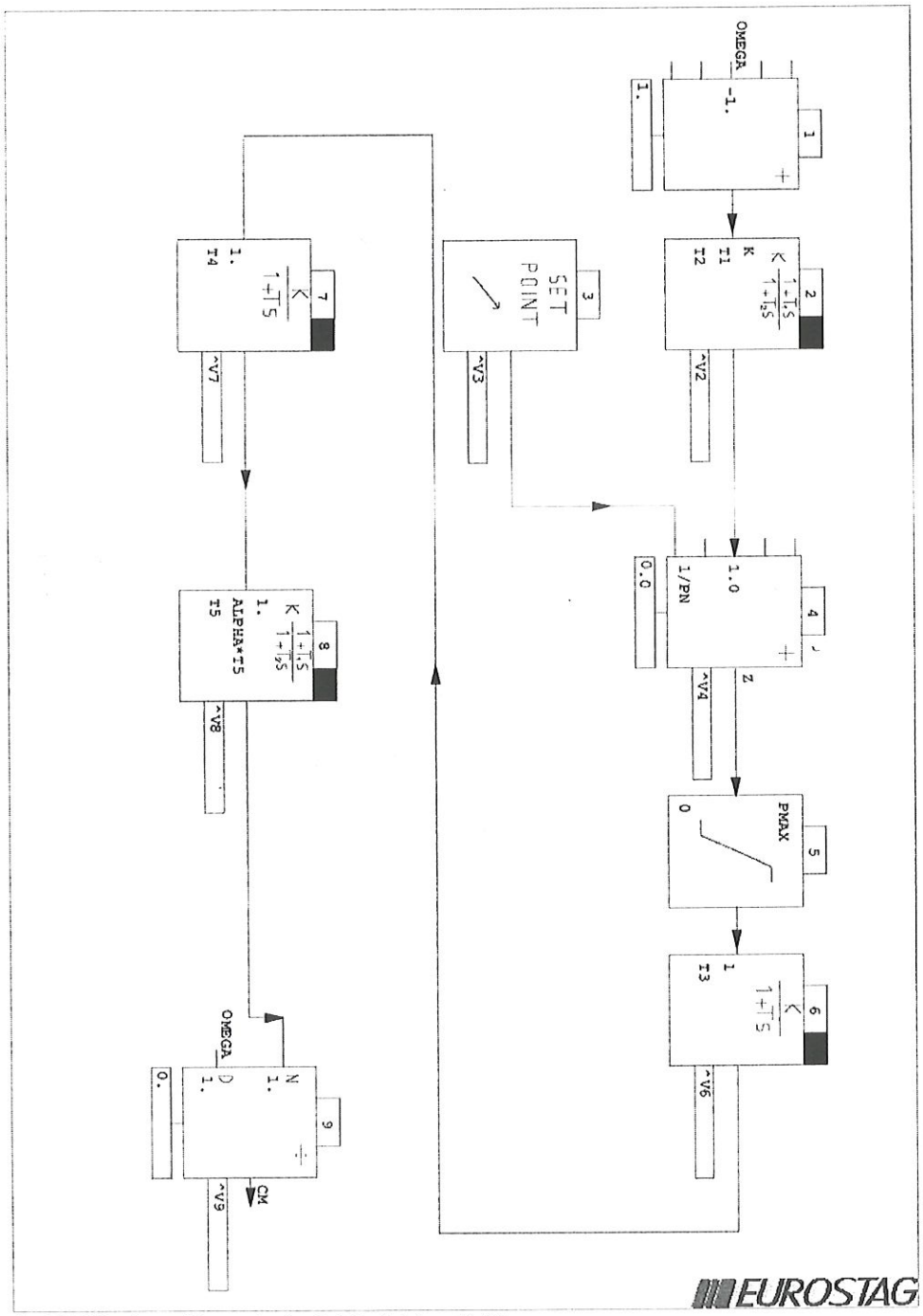

FIG. 2.17 – La régulation gover3

une variable d'interface. Ensuite, du bloc 6 au bloc 8, nous avons, à chaque bloc, une nouvelle équation car ce sont des blocs différentiels dont la sortie est une variable d'état. Nous obtenons enfin une dernière équation car la sortie du macrobloc est définie comme étant une variable d'interface. Nous allons noter y_i la variable d'état du bloc i. Nous avons alors comme équations :

1. $\dot{y_2} = -\frac{y_2}{T2} + \frac{KT1}{T2}(1 - \dot{OMEGA}) + \frac{K}{T2}(1 - OMEGA)$
2. $z = y_2 + \frac{y_3}{PN}$
3. $\dot{y_6} = \frac{1}{T3}z - \frac{1}{T3}y_6$
4. $\dot{y_7} = \frac{1}{T4}y_6 - \frac{1}{T4}y_7$
5. $\dot{y_8} = -\frac{y_8}{T5} + ALPHA\dot{y_7} + \frac{1}{T5}y_7$
6. $\frac{y_8}{OMEGA} = y_9$
7. $y_9 = CM,$

ce qui correspond à un système avec 4 équations différentielles et 3 équations algébriques.

Reprenons les équations ci-dessus et écrivons les au temps initial. Nous allons prendre comme convention que la valeur initiale du bloc i sera Vi.

Première équation : On remplace d'abord les dérivées par zéro, le membre de gauche et le deuxième terme du membre de droite s'annulent alors. L'équation devient :

$$0 = -\frac{V2}{T2} + \frac{K}{T2}(1 - OMEGA_0).$$

En multipliant par $T2$ et en arrangeant les termes, l'équation devient alors :

$$V_2 = K(1 - OMEGA_0).$$

On peut remarquer que cela revient à remplacer dans le macrobloc le bloc par un bloc multiplicateur.

Deuxième équation : Comme il n'y a pas de dérivée dans cette équation, elle ne change pas, on évalue juste les variables en zéro. On aura donc comme équation :

$$V4 = V2 + \frac{V3}{PN}.$$

Troisième équation : On remplace les dérivées par zéro, on multiplie par $T3$ et on arrange les termes. L'équation devient alors :

$$V_6 = V_4.$$

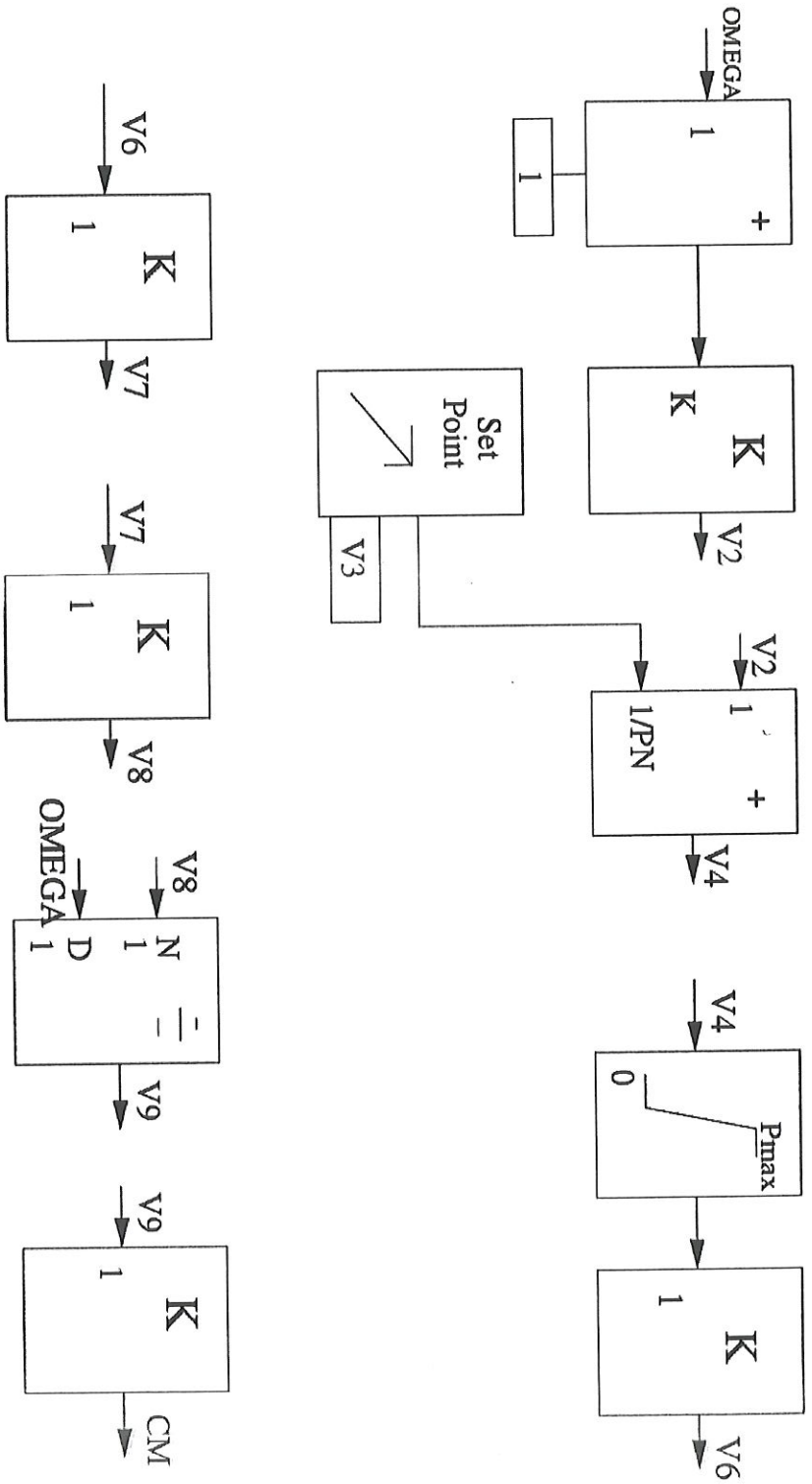

FIG. 2.18 – La régulation gover3 séprée en équations avec les blocs remplacés

- Cela nous permet de résoudre l'équation 5 en la remontant (car il faut trouver une entrée et non la sortie). On connaît donc V7.

- Ensuite, on résoud l'équation 4 en la remontant (car il faut trouver une entrée et non la sortie). On connaît donc V6.

- Puis, on résoud l'équation 3 en la remontant (car il faut trouver une entrée et non la sortie). On connaît donc V4.

- On résoud enfin l'équation 2 en la remontant (car il faut trouver une entrée et non la sortie). On connaît donc V3.

- Toutes les valeurs initiales sont connues, on a terminé.

Remarquons que cet exemple est très simple à résoudre car il ne contient aucune alternative. Cependant, il est très rare dans la réalité d'avoir des régulations sans aucune alternative. Les alternatives feront l'objet du prochain chapitre.

Chapitre 3

Traitement des blocs avec alternatives

3.1 Explication de la notion d'alternatives

Contrairement aux blocs simples, les blocs dits *avec alternatives* ont la caractéristique importante d'avoir plusieurs équations différentes. En effet, lors de la descente de l'équation, la valeur de l'entrée déterminera l'équation à choisir. Pour expliquer cette notion, prenons l'exemple du bloc *limiter*. Sa représentation graphique est présentée à la figure 3.1 et l'équation correspondant au bloc est l'équation (3.1).

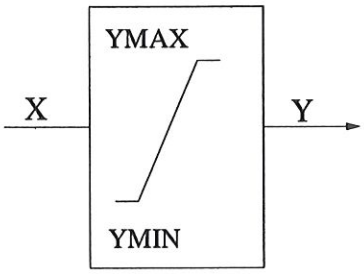

FIG. 3.1 – Le bloc limiteur

$$\begin{cases} y = x & \text{si } YMIN < x < YMAX \\ = YMAX & \text{si } x \geq YMAX \\ = YMIN & \text{si } x \leq YMIN \end{cases} \tag{3.1}$$

On voit bien que la fonction du bloc est de limiter l'entrée à un certain intervalle donné et que selon la valeur de l'entrée l'équation est différente.

Nous venons d'observer un bloc avec alternatives, c'est-à-dire un bloc dont l'équation change selon la valeur de l'entrée, mais il y a aussi des blocs que nous traiterons d'une manière similaire, *les blocs algébriques sans inverse*. En effet, nous pouvons supposer que ces blocs se comportent comme des blocs avec alternatives lors de la remontée du bloc. Selon la valeur de la sortie, on aura une équation différente pour déterminer l'entrée. Nous avons représenté le bloc *abs* à la figure 3.2 et qui correspond à la fonction valeur absolue, $y = |x|$. Nous voyons bien que lors de la remontée, il y a un dilemne. Supposons que $y = 2$, nous ne saurons pas si $x = 2$ ou si $x = -2$. C'est pour cela qu'il seront traités de la même manière que les blocs avec alternatives.

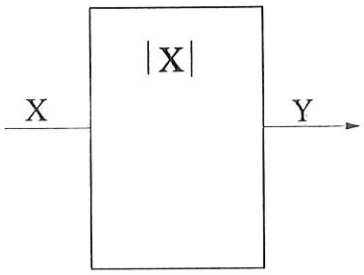

FIG. 3.2 – Le bloc valeur absolue

Il y a aussi des blocs différentiels avec alternatives. Nous avons représenté à la figure 3.3 le bloc *limited simple lag*. Son équation se trouve à l'équation (3.2).

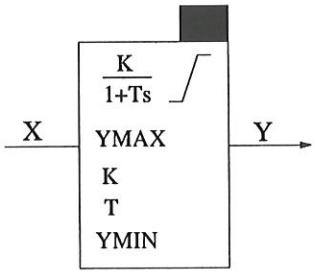

FIG. 3.3 – Le bloc limited simple lag

$$\begin{cases} \dot{y} = -\frac{1}{T}y + \frac{K}{T}x & \text{si} \begin{cases} YMIN < y < YMAX \\ \text{ou } y = YMIN \text{ et } -\frac{1}{T}y + \frac{K}{T}x > 0 \\ \text{ou } y = YMAX \text{ et } -\frac{1}{T}y + \frac{K}{T}x < 0 \end{cases} \\ = 0 \qquad \text{sinon} \end{cases} \tag{3.2}$$

On peut tout de suite imaginer la difficulté que peuvent entraîner ces blocs pour la résolution d'équations. En effet, supposons que l'entrée d'un tel bloc dépende d'une valeur initiale encore non déterminée, il sera impossible de connaître exactement cette équation, il faudra faire différentes hypothèses, ce qui amènera à des équations différentes.

3.2 Analyse de deux blocs représentatifs

Nous allons à présent analyser deux blocs avec alternatives, un algébrique et un différentiel. Nous avons choisi d'analyser le bloc limiter car c'est un des blocs les plus répandus et parce qu'il est assez simple à imaginer. Le bloc limiter restreint l'entrée à un certain intervalle. Cela permet, par exemple, d'éviter que la sortie du bloc devienne beaucoup trop grande. Sa représentation graphique peut être trouvée à la figure 3.1 et son équation en (3.1).

La descente du bloc peut toujours s'effectuer mais pas la remontée. En effet, il est impossible que la sortie du bloc ne soit pas dans l'intervalle et on ne pourra dès lors pas remonter le bloc si c'est le cas. On voit également que si la sortie est une des deux bornes de l'intervalle, il est alors impossible de déterminer exactement l'entrée.

Nous reprenons le même schéma d'analyse que celui pour les blocs simples. Voici donc l'analyse du bloc limiter :

Description Ce bloc permet de limiter la sortie à un intervalle $[ymin, ymax]$.

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut remonter ce bloc uniquement si $y(0) \in [ymin, ymax]$.

Résolution

- Si on descend le bloc, on restreint l'entrée a l'intervalle.

- Si on remonte le bloc, il y a plusieurs cas :
  1. Si $y(0) \in]ymin, ymax[$, alors on a $x(0) = y(0)$.
  2. Si $y(0) = ymax$, on sait juste que $x(0) \geq ymax$.
  3. Si $y(0) = ymin$, on sait juste que $x(0) \leq ymin$.
  4. Si $y(0) \notin [ymin, ymax]$, il faut renvoyer une erreur.

Nous avons choisi d'analyser le bloc *Limited simple lag* pour illustrer les blocs différentiels avec alternatives. Sa représentation graphique peut être trouvée à la figure 3.3 et son équation en (3.2).

Nous obtenons l'analyse suivante :

Description C'est un bloc Simple Lag dont on a limité la sortie :

$$\begin{array}{rcll} y(0) & = & Kx(0) & \text{si} \quad ymin < y < ymax \\ & & & \text{si} \quad y = ymin \text{ et } Kx \geq ymin \\ & & & \text{si} \quad y = ymax \text{ et } Kx \leq ymax \\ & = & 0 & \text{sinon} \end{array}$$

Valeur Initiale ? Oui, $y(0)$

Remplacement Oui, par un bloc Gain avec un paramètre K suivi d'un bloc limiter de paramètre $ymin$ et $ymax$.

Sens

- On peut toujours descendre ce bloc.
- On peut remonter ce bloc uniquement si $y(0) \in [ymin, ymax]$.

Résolution Il y a plusieurs cas à envisager :

- Si on descend le bloc, il y a trois cas :
  1. Si $Kx(0) < ymin$, alors $y(0) = ymin$
  2. Si $Kx(0) > ymax$, alors $y(0) = ymax$
  3. Sinon $y(0) = Kx(0)$.

- Si on remonte le bloc,
  1. Si $y(0) \notin [ymin, ymax]$, alors il faut renvoyer une erreur.
  2. Si $y(0) = ymin$, alors on sait juste que $Kx(0) < 0$.
  3. Si $y(0) = ymax$, alors on sait juste que $Kx(0) > 0$.
  4. Sinon $y(0) = Kx(0)$.

3.3 Notion d'arborescence déduite

Nous allons représenter une équation contenant plusieurs blocs avec alternatives et nous allons pour cela faire l'analogie avec une arborescence. Chaque bloc rencontré sera un noeud de cette arboresence. Il y aura une seule branche vers le noeud suivant si le bloc est simple mais il y en aura autant que d'alternatives si le bloc est avec alternatives,

chaque branche représentant alors la suite des blocs en faisant l'hypothèse que l'alternative est vérifiée.

Illustrons cette notion avec un exemple. Supposons que nous ayons une équation composée d'un bloc simple suivi d'un bloc avec 3 alternatives, d'un bloc simple et enfin d'un bloc avec 3 alternatives. On obtient alors la structure de la figure 3.4.

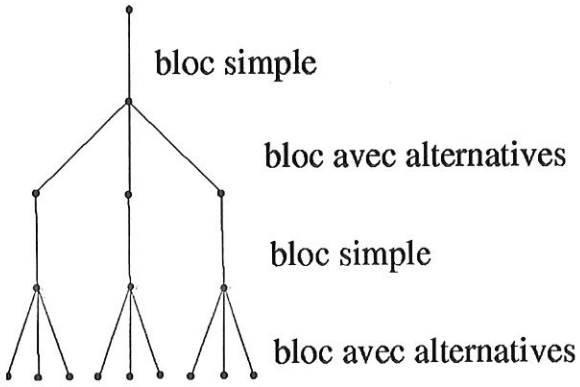

FIG. 3.4 – Structure d'arborescence avec deux blocs à alternatives

Nous avons, à chaque fois, supposé qu'il y avait 3 alternatives, car c'est ce qui arrive le plus souvent. Chaque feuille de l'arbre est une configuration possible de la machine. En effet, chaque feuille correspond à une équation différente qui fera donc partie d'un système d'équations différent, ce qui mènera à des solutions différentes, donc à des valeurs initiales différentes. Il est possible que des alternatives soient incompatibles ou que le système d'équations ne possède pas de solution, cela représentera des états de la machine qui ne sont pas possibles. Il faudra alors dans ce cas éliminer la feuille et donc la branche correspondante.

Nous pouvons voir comme le nombre d'états augmente exponentiellement avec le nombre de blocs avec alternatives. Dans cet exemple très simple, nous obtenons déjà 9 configurations différentes pour seulement deux blocs avec 3 alternatives chacun.

Nous pouvons également représenter le macrobloc par une arborescence. En effet, maintenant que nous connaissons le nombre de configurations différentes possibles pour chaque équation, nous pouvons les combiner afin d'obtenir le nombre de configurations totales du macrobloc. Supposons que le macrobloc soit constitué de 4 équations (ce qui est très peu). La première équation n'est composée que de blocs simples, la deuxième équation est celle traitée ci-dessus, la troisième équation est simple également et la dernière n'est composée que d'un seul bloc avec 3 alternatives. On a alors l'arborescence de la figure 3.5.

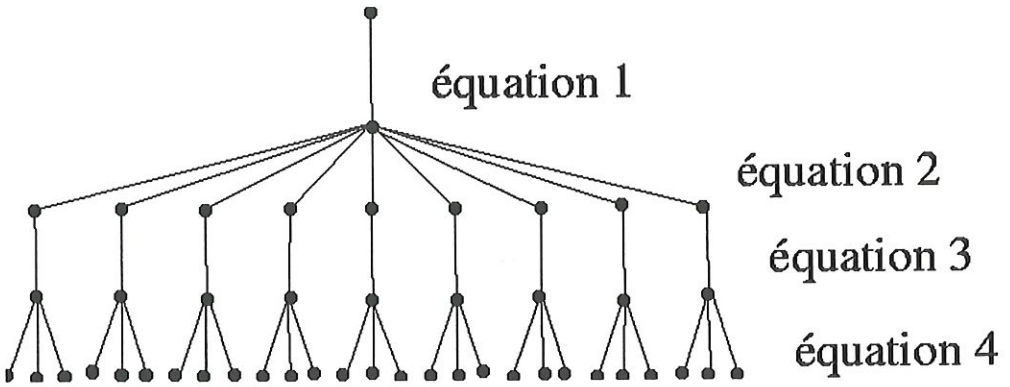

FIG. 3.5 – Structure d'arboresence pour quatre équations

Nous pouvons constater que, même pour un macrobloc simple, le nombre de configurations est énorme. En effet, ce macrobloc n'est composé que de 4 équations et de 3 blocs avec alternatives mais comporte 27 configurations. Il va de soi qu'il est impensable de résoudre 27 systèmes non linéaires et ensuite trouver la meilleure solution pour deux raisons. Premièrement, cela prendrait beaucoup trop de temps. Ensuite, on ne sait pas déterminer quelle solution est la meilleure. Une machine démarre 99.9% du temps dans un état particulier qu'on appellera *l'état normal de démarrage*. Cet état correspond à une des feuilles et le but du programme est de trouver cet état particulier mais il n'existe aucune fonction "objectif" permettant de préférer cette solution à une autre. Nous allons donc construire dans la section suivante une méthode permettant d'éliminer au plus vite les hypothèses inutiles afin d'arriver au plus vite aux solutions.

3.4 Proposition d'une solution

Après avoir séparé le macrobloc en plusieurs groupes de blocs correspondant chacun à une équation du macrobloc et remplacé les blocs élementaires par leur équivalent au temps initial, il faut maintenant trouver les valeurs initiales des variables d'état. On va essayer d'éliminer le plus rapidement possible des alternatives afin de diminuer le travail.

Chaque groupe de blocs forme un système ayant :

- une équation
- de zéro à m inéquations.

En effet, on a en général une équation mais quand on a un bloc avec alternatives et lorsqu'on fait l'hypothèse qu'une des alternatives est vérifiée, cela implique une ou plusieurs inégalités.

La figure 3.6 montre un groupe de blocs représentant uniquement une équation sans aucune inéquation. L'équation représentée est $x = cos(e^2)$.

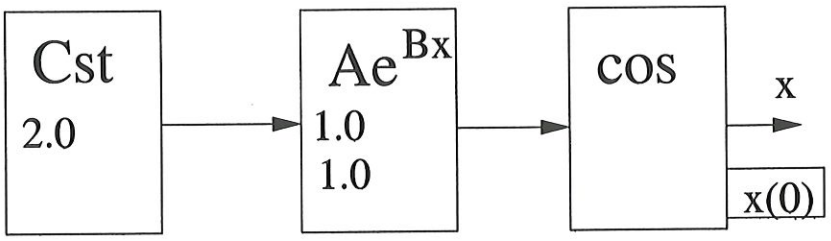

FIG. 3.6 – Présentation d'une équation

La figure 3.7 montre un groupe de blocs représentant une équation ainsi qu'une ou deux inéquations selon le cas. En effet, on a plusieurs cas à envisager :

- Si on suppose que l'entrée du bloc limiter est plus petite que $-a$, on aura comme équation $x = cos(-a)$ et comme inéquation $2.0 \leq -a$.

- Par contre, si on suppose que l'entrée du bloc limiter est plus grande que a, on aura comme équation $x = cos(a)$ et comme inéquation $2.0 \geq a$.

- Enfin, si on suppose que l'entrée du bloc limiter est comprise entre $-a$ et a, on aura comme équation $x = cos(2.0)$ et comme inéquations $-a < 2.0$ et $2.0 < a$.

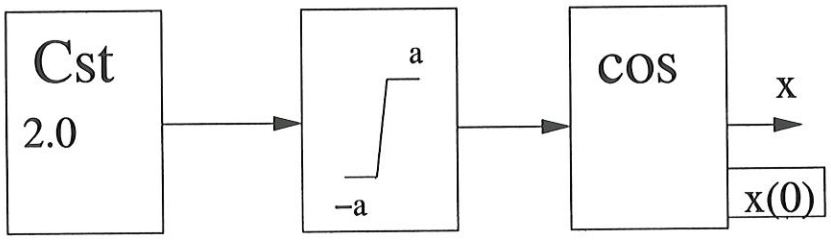

FIG. 3.7 – Présentation d'une équation

Par simplicité, on désignera dorénavant par équation le groupe de blocs formant les équations et les inéquations lorsqu'il n'y a pas d'ambiguité. C'est à dire que l'on désignera par équation l'ensemble des blocs et pas l'équation déduite en évaluant les blocs.

A chaque équation, on va associer un poids qui sera, au départ, le nombre de degrés de liberté de cette équation (c'est-à-dire le nombre de valeurs initiales qui sont encore

totalement libres). Ensuite, le poids de chaque équation encore non traitée sera diminué afin de représenter la corrélation de cette équation avec les équations déjà traitées. Plus le poids sera petit, plus la corrélation sera grande. Nous permettons que ce poids devienne négatif. En effet, il se peut que la corrélation entre les équations deviennent tellement forte que les poids deviendront négatifs et nous ne l'empêchons pas.

Reprenons, par exemple, le schéma initial de la régulation gover3. Pour rappel, le schéma initial est composé des équations :

1. $K(1 - OMEGA_0) = V2$
2. V4=V2+$\frac{V3}{PN}$
3. V6=V4
4. V7=V6
5. V8=V7
6. $\frac{V8}{OMEGA_0} = V9$
7. V9=CM_0.

On peut alors associer comme poids aux équations le vecteur $(1, 3, 2, 2, 2, 2, 1)^T$. En effet, la première équation a un seul degré de liberté, la deuxième en a trois, la troisième deux, etc. Ensuite, lors de la résolution du système, on traite les équations 1 et 7 (qui sont les équations de poids minimal). Les variables V2 et V9 sont alors connues et en posant ces variables comme connues, on obtient alors comme vecteur de poids $(0, 2, 2, 2, 2, 1, 0)$. Et on peut continuer la résolution en traitant l'équation 6.

On va commencer par traiter l'équation de poids minimal puisque cette équation est celle qui est la plus simple à résoudre et permettra même, si elle a un degré de liberté de un, de déterminer une valeur initiale. Il se peut que l'équation contienne un bloc avec alternatives. On devra dès lors traiter chaque alternative séparément : on supposera qu'une des alternatives est vraie et on regarde si cela est possible avec le reste des équations. Nous parlerons alors d'*hypothèses d'une équation.*

Si l'équation possède plusieurs blocs avec alternatives, il faudra envisager, pour chaque hypothèse, d'autres hypothèses que nous appelerons *sous-hypothèses.* Nous parlerons de sous-hypothèses de niveau 1 pour la première sous-hypothèse envisagée, de niveau 2 pour la deuxième, etc. Ces sous-hypothèses sont à traiter avant les autres équations.

La figure 3.7 est une équation possédant 3 hypothèses. Tandis que la figure 3.8 possède 3 hypothèses ainsi qu'un niveau de sous-hypothèses contenant 3 hypothèses. En effet, les hypothèses seront les trois alternatives du premier bloc rencontré ($V1 < -b$, $V1 > b$ et $-b \leq V1 \leq b$). Les sous-hypothèses correspondront aux trois alternatives du deuxième bloc rencontré (si on appelle x, l'entrée du deuxième bloc, les sous-hypothèses

seont $x < -a$, $x > a$ et $-a \leq x \leq a$). Cependant, certaines hypothèses contrediront les sous-hypothèses et on aura donc des branches de la représentation en arbre qui seront à couper. Pour savoir si les hypothèses se contredisent, il faudra regarder si l'ensemble défini par ces hypothèses est admissible. S'il ne l'est pas, la branche sera à couper. Par exemple, dans l'exemple présenté à la figure 3.8, si on suppose qua $a > 0$, $b > 0$ et que $a < b$, alors, par exemple, si on envisage l'hypothèse $V1 < -b$, la sortie du premier bloc sera égale à $-b$ et la sous-hypothèse $-b > a$ sera impossible.

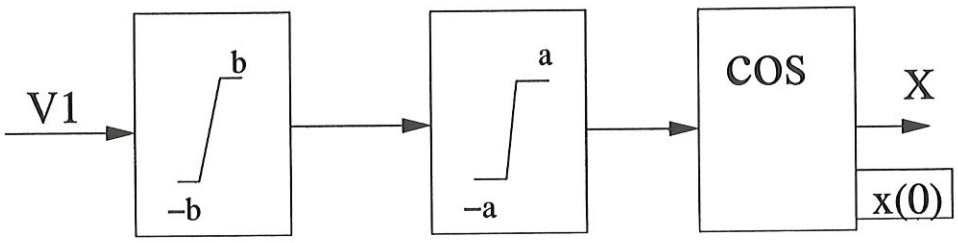

FIG. 3.8 – Présentation d'une équation

Nous allons maintenant expliquer comment la méthode déduite traite un problème. On envisage la première hypothèse de l'équation de poids minimal. Il y a alors plusieurs cas à envisager :

- Si l'ensemble déterminé par l'équation est admissible alors :

  1. S'il n'y a plus d'autres équations et plus de sous-hypothèses à traiter, cela veut dire que l'on avait un macrobloc avec une seule équation. On prendra alors un point admissible qui sera une solution du problème et,

     - s'il n'y a plus d'hypothèses, on a une seule équation qui n'a aucun bloc avec alternatives. On vient de traiter cette équation, on a donc fini l'algorithme.

     - sinon, il y a une seule équation mais celle-ci a plusieurs hypothèses, on passe alors à l'hypothèse suivante.

  2. S'il y a d'autres équations ou des sous-hypothèses, alors on va devoir déterminer la deuxième équation et la traiter en sachant que l'ensemble admissible a été restreint puisqu'on y a ajouté l'équation traitée. On traitera d'abord les sous-hypothèses de l'équation et ensuite on envisagera les autres équations.

- Si l'ensemble n'est pas admissible, cela veut dire que l'équation n'a pas de solution, alors :

1. S'il n'y a plus d'autres hypothèses, on a fini l'algorithme.

2. Sinon on passe à l'hypothèse suivante.

On remet alors à jour les degrés de liberté des équations encore non traitées étant donné qu'on a eu une information sur certaines variables.

Si on doit envisager la deuxième hypothèse de l'équation, le même raisonnement est à refaire.

Sinon s'il y a des sous-hypothèses à envisager, alors on les envisage. Sinon on choisira l'équation de poids minimal qui sera soit une équation très simple ou alors une équation fortement correlée avec la précédente ce qui permettra d'éliminer rapidement des alternatives. Pour traiter la deuxième équation ou les sous-hypothèses, on fera :

- Si l'ensemble déterminé par l'équation et par les équations précédentes est admissible, alors :

  1. S'il n'y a plus d'autres équations et plus de sous-hypothèses à traiter, on prendra un point admissible qui sera une solution du problème et,

     - s'il n'y a plus d'hypothèses, la régulation a uniquement deux équations. On a traîté la deuxième équation et on doit maintenant traîter l'hypothèse suivante de la première équation.

     - sinon on passe à l'hypothèse suivante de cette équation.

  2. S'il y a d'autres équations ou des sous-hypothèses, alors on va devoir déterminer la troisième équation et la traiter en sachant que l'ensemble admissible a été restreint.

- Si l'ensemble n'est pas admissible alors

  1. Il y a une incompatibilité des hypothèses. On doit donc retourner à l'équation précédente et nous traiterons l'hypothèse suivante.

  2. Sinon on passe à l'hypothèse suivante.

On remet à jour les degrés de liberté et on continue ainsi jusqu'à avoir traité toutes les équations.

On obtient alors l'algorithme suivant :

Algorithme :

Si on a traité toutes les hypothèses de l'équation, alors :

1. Si on est à la première équation, on a terminé l'algorithme.
2. Sinon, on passe à l'hypothèse suivante de l'équation précédente avec un ensemble admissible dont les équations et inéquations de l'hypothèse en cours de l'équation précédente ont été enlevées.

sinon :

1. Si l'ensemble admissible auquel on ajoute l'ensemble des équations et inéquations de l'hypothèse en cours de l'équation en cours est admissible, alors
   - Si on a traité toutes les équations et sous-hypothèses, alors on recherche un point admissible qui sera une de nos solutions et on passe à l'hypothèse suivante de cette équation.
   - Sinon on doit passer à l'équation suivante. Celle-ci sera soit les sous-hypothèses de l'équation s'il y en a ou l'équation suivante de poids minimal.
2. Si l'ensemble n'est pas admissible, on doit alors passer à l'hypothèse suivante de l'équation en cours.

Si on suppose que les ensembles sont toujours admissibles, on obtient alors l'ordre de résolution présenté à la figure 3.9. On peut avoir une solution uniquement si on se trouve sur une feuille de l'arbre. L'arbre représenté est assez simple mais il se peut que l'arbre ne soit pas du tout équilibré du fait que l'on peut ou non avoir des sous-hypothèses selon le choix de la première hypothèse. Dans l'exemple représenté à la figure 3.9, on a neuf solutions, les numéros 5, 6, 7, 10, 11, 12, 15, 16 et 17.

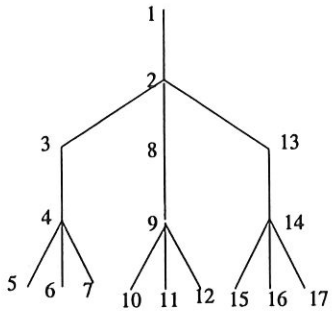

FIG. 3.9 – Explication du chemin effectué par la méthode

Si on suppose à présent que l'ensemble n'est pas toujours admissible, on va représenter à la figure 3.10 le fonctionnement de la méthode. Les X représentés sur la figure seront les endroits où l'ensemble des équations n'est pas admissible. La figure représentée a deux solutions, les numéros 4 et 6.

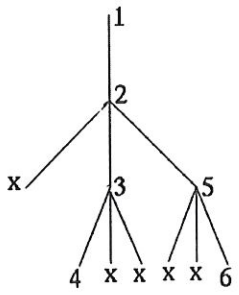

FIG. 3.10 – Explication du chemin effectué par la méthode

Cette méthode correspond à une méthode branch-and-bound dont on choisit l'équation suivante comme celle qui a le poids le plus petit, c'est-à-dire celle qui apporte le plus d'informations, soit en donnant la valeur d'une variable, soit en essayant de trouver une équation fortement correlée aux précédentes.

Cependant, cette méthode ne nous convient pas car elle va nous donner toutes les solutions possibles et ce sera à l'ingénieur de dire celle qu'il lui faut. Nous allons donc essayer de trouver une méthode générale de résolution qui, en lui donnant quelques hypothèses supplémentaires, va trouver l'unique état de la machine qui nous convient.

De plus, l'évaluation de l'alternative nécessite la connaissance des valeurs des variables d'interface prédéfinies, s'il y en a. Cependant, ces valeurs ne sont connues que lors de la simulation. Le module de création du schéma initial est utilisé de façon déconnectée du module de simulation et avant celui-ci (comme présenté dans la présentation

du logiciel Eurostag dans le premier chapitre). Il est impensable de mettre cette initialisation automatique en début de simulation. En effet, l'utilisateur fait plusieurs dizaines de simulations par jour et chaque simulation contient plusieurs centaines de macroblocs à initialiser et on ne peut dès lors demander quoi que ce soit à l'utilisateur lors de la simulation. Il faut donc laisser le module d'initialisation automatique avant le module de simulation et initialiser les blocs une fois pour toutes et trouver un moyen de faire des choix automatiquement. Ce sera l'objet du chapitre suivant.

Chapitre 4

Méthode générale de résolution

4.1 Hypothèses supplémentaires

Nous allons faire l'hypothèse que, pour les blocs qui limitent la sortie, on fait comme s'il n'y avait pas de limite sauf si l'ingénieur précise le contraire. Par exemple, nous considérons le bloc *simple lag limité* comme un bloc *simple lag* "normal" si l'ingénieur n'a rien précisé. Pour l'instant, avec la version actuelle du logiciel Eurostag, il n'est pas possible de préciser quelle alternative choisir "par défaut" à l'initialisation. Cependant, la plupart du temps, l'ingénieur connaît cette alternative et le programme devra être modifié afin d'ajouter un menu contextuel permettant de dire quelle alternative doit être choisie.

Une même hypothèse pourra être faite pour d'autres blocs avec alternatives qui ne correspondent pas à des blocs limités et on pourra même, pour certains blocs, se passer du menu contextuel. Par exemple, on pourra supposer que les entrées d'un bloc minimum sont classées de haut en bas par ordre croissant et on sélectionnera donc, par défaut, la première entrée.

Pour les autres blocs avec alternatives où il n'est pas possible de faire une hypothèse simple, comme les blocs algébriques simples dont la fonction n'est pas inversible, il peut s'avérer assez lourd de demander à l'utilisateur, pour chaque bloc, l'alternative à choisir. Nous n'avons pas trouver de moyens efficaces et moins lourd pour traîter ces blocs et donc, dans notre approche théorique du problème, on suppose qu'on connaît l'alternative à choisir.

Il faut noter que l'hypothèse sur les blocs limitateurs n'est pas trop forte. En effet, ces blocs permettent surtout de réguler la machine, de l'empêcher d'aller trop vite, d'être en surtension, etc, et puisque les systèmes électriques sont supposés stationnaires au temps initial, il est peu probable qu'une machine se trouve dans un état pareil. Cependant, un ingénieur peut imposer un état de démarrage quelconque à la machine et c'est pourquoi nous laissons la liberté de fixer l'alternative.

Il y a un bloc avec alternatives à traiter différemment des autres, le bloc *relay*. Il est représenté à la figure 4.1 et son équation se trouve à l'équation (4.1). En effet, ce bloc choisit, selon le signe de la première entrée, si la sortie est égale à la deuxième ou à la troisième entrée. Nous sommes donc obligés de connaître la première entrée avant d'envisager l'équation dont le relay fait partie. Nous aurons donc deux possibilités. Soit l'entrée est constante ou peut être déterminée facilement avant d'envisager l'équation du relay. Soit il faudra renvoyer un message à l'utilisateur en lui demandant de donner la valeur de toutes les valeurs initiales faisant partie de la première entrée du relay.

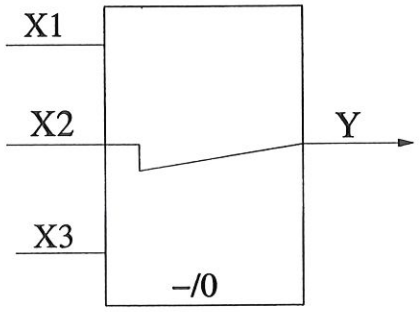

FIG. 4.1 – Le bloc relay

$$y = \begin{cases} x2 & \text{si} \quad x1 > 0 \\ x3 & \text{si} \quad x1 \leq 0 \end{cases} \tag{4.1}$$

En fixant des alternatives, on ne va pas tenir compte de certaines parties du macrobloc. Mais si des valeurs initiales sont présentes dans les morceaux dont on ne tient pas compte, il faudra tout de même leur donner une valeur initiale puisque ces valeurs seront utilisées par la suite dans la simulation. Il faudra donc demander à l'utilisateur de donner cette valeur initiale.

De plus, en fixant des alternatives, on obtient une équation mais aussi une ou plusieurs inéquations. Ceci va donc nous donner le système électrique qui est composée des équations ainsi que toute une série d'inéquations. Nous savons que les systèmes d'équations non linéaires peuvent ne pas avoir de solution, en avoir une seule ou même avoir plusieurs solutions. Nous allons cependant supposer que, sur le domaine défini par les inéquations, le système a une solution unique.

La dernière hypothèse à formuler est qu'on dispose d'un outil permettant de résoudre les équations. En effet, il est peu probable de pouvoir résoudre toutes les équations simplement en remplaçant les variables connues dans les équations encore non-traitées.

Nous allons être amenés à résoudre des systèmes d'équations couplées et nous utiliserons une méthodes de résolution d'équations non-linéaires avec contraintes d'inégalités. Le programme Eurostag ne permet pas encore de résoudre de tels systèmes.

4.2 Méthode déduite

Après avoir séparé le macrobloc en équations, remplacé les blocs par leur équivalent au temps initial, remplacé les variables d'interface prédéfinies par leur valeur, il nous reste un système de m équations à n inconnues.

- Si $m < n$, nous allons devoir renvoyer un message à l'utilisateur en lui demandant de fixer $m - n$ variables. Nous pouvons donc supposer qu'il y a plus ou autant d'équations que d'inconnues.

- Le programme d'analyse topologique présent dans le troisième module d'Eurostag empêche que $m > n$. On pourra donc supposer qu'il y a toujours autant d'inconnues que d'équations.

Nous allons tout d'abord résoudre les équations à une variable. Ces équations permettent de trouver une valeur initiale et nous continuerons à chercher et à résoudre des équations à une variable en flaggant les valeurs initiales comme connues jusqu'à ce qu'il ne soit plus possible d'en trouver. Les équations sont à résoudre dans un ordre quelconque.

Ensuite, il faut fixer les alternatives. En effet, cette phase va avoir lieu seulement à ce point car, ainsi, on espère pourvoir trouver des alternatives sans avoir à demander quoi que ce soit à l'utilisateur. On voit bien que si l'entrée d'un bloc avec alternatives n'est composée que de variables qui peuvent être déterminées en résolvant les équations à une inconnue, on ne doit alors pas demander à l'utilisateur de fixer l'alternative.

A ce point, il y a deux possibilités :

- Nous avons traité toutes les équations et trouvé toutes les valeurs initiales. Nous avons donc terminé l'algorithme.

- Il reste des équations et des valeurs initiales à trouver.

C'est le deuxième point qui nous intéresse. On va alors devoir trouver un sytème de p équations à p inconnues à résoudre avec p le plus petit possible. Supposons que nous ayons trouvé ce système, nous le résolvons grâce à notre outil de résolution et ensuite nous recherchons et résolvons les équations à une inconnue en supposant que toutes les valeurs initiales intervenant dans le système soient connues et nous répétons ce procédé

jusqu'à ce que toutes les valeurs initiales aient été trouvées.

Il nous reste donc à savoir comment trouver ce système. Le but est de trouver le plus petit système possible. En effet, il est beaucoup plus rapide de résoudre des équations à une variable et nous allons donc essayer de revenir le plus vite possible à ce procédé. Nous allons déterminer la première équation du système comme l'équation ayant le moins de degrés de liberté. S'il y en a plusieurs, on fera le raisonnement pour chacune d'elles et, ensuite, une fois chaque système déterminé, on choisira le plus petit système. Ce raisonnement peut être un peu lent mais permettra d'obtenir le plus petit système possible. La deuxième équation du système sera l'équation ayant le plus d'inconnues en commun avec la première et de poids minimal et nous déterminerons la troisième équation comme étant celle qui a le plus de variables en commun avec les deux premières et de poids minimal et ainsi de suite jusqu'à ce que le nombre d'inconnues soit égal au nombre d'équations. Comme le nombre d'inconnues est plus petit que le nombre d'équations, il sera toujours possible de trouver un tel système.

Voici donc l'algorithme de résolution déduit :

- Tant qu'il y a des équations avec une seule variable inconnue faire
  - Déterminer la variable inconnue
  - Marquer la variable comme connue
- Demander à l'utilisateur de fixer les alternatives qui ne peuvent être déterminées.
- Tant que toutes les équations ne sont pas traitées faire
  - Tant qu'il y a des équations avec une seule variable inconnue faire
    - Déterminer la variable inconnue
    - Marquer la variable comme connue
  - S'il reste encore des équations à traiter
    - Alors déterminer un système d'équations couplées, le résoudre et marquer les variables trouvées comme connues.

Et voici l'algorithme permettant de trouver le système :

- Trouver les équations ayant le moins de degrés de liberté
- Pour chaque équation trouvée, faire
    - Rajouter cette équation au système
    - Tant que le nombre d'équations est plus petit que le nombre d'inconnues faire
        - Trouver l'équation ayant le plus de variables en commun avec les équations du système de poids minimal
        - Rajouter cette équation au système
- Choisir le système ayant le moins d'équations.

4.3 Application de la méthode à la régulation gover3

Nous allons appliquer l'algorithme trouvé à la régulation gover3 afin de montrer que l'algorithme fonctionne bien sur un exemple simple sans aucune alternative. Le macrobloc de la régulation se trouve à la figure 2.17. Comme déjà fait dans le chapitre deux, en séparant le macrobloc en équations et en remplaçant les blocs différentiels par leur équivalent au temps initial, nous trouvons les équations suivantes :

1. $K(1 - OMEGA_0) = V2$
2. V4=V2+$\frac{V3}{PN}$
3. V6=V4
4. V7=V6
5. V8=V7
6. $\frac{V8}{OMEGA_0} = V9$
7. V9=CM_0.

L'algorithme va alors résoudre les équations dans le même ordre que celui qu'on avait trouvé précédemment. Cependant, nous avons toujours trouvé des équations à une inconnue et nous n'avons pas eu besoin de déterminer un système. Nous allons donc modifier les équations afin de pouvoir tester la deuxième partie de l'algorithme.

4.4 Application de la méthode à la régulation gover3 modifiée

Nous modifions les équations 1 et 5 afin de tester notre méthode. Le système devient alors :

1. $K(1 - OMEGA_0) = V2 + V3$
2. V4=V2+$\frac{V3}{PN}$
3. V6=V4
4. V7=V6
5. V8=V7 + V6
6. $\frac{V8}{OMEGA_0} = V9$
7. V9=CM_0.

L'application de l'algorithme va s'effectuer dans l'ordre suivant :

- On résoud d'abord l'équation 7. On marque la variable V9 comme connue.
- On résoud ensuite l'équation 6. On marque la variable V8 comme connue.
- Il n'y a alors plus d'équations à une inconnue. On recherche alors un système d'équations. On cherche les équations de poids minimal. Il y en a quatre, les équations 1, 3, 4 et 5. En effet, elles ont chacune deux degrés de liberté. Pour chaque équation, on va rechercher le système associé :
  - Le système correspondant à l'équation 1 comprend 5 équations. En effet, le système comprend l'équation 1. On recherche l'équation ayant le plus de variables en commun. On ajoute donc l'équation 2 au système. Ensuite, on ajoute l'équation 3 puis l'équation 4 et puis la 5.
  - Le système correspondant à l'équation 3 comprend 3 équations. En effet, le système comprend l'équation 3. On ajoute ensuite l'équation ayant le plus de variables en commun avec l'équation, l'équation 4 et ensuite on ajoute l'équation 5 car c'est l'équation qui a le plus de variables en commun avec les deux précédentes.
  - Le système de l'équation 4 comprend 2 équations, l'équation 4 et l'équation 5 car c'est celle qui a le plus de variables en commun.

- Le système de l'équation 5 comprend 2 équations également, les mêmes que pour le système de l'équation 4.

Nous allons donc choisir le système de l'équation 4. Il comprend les équations 4 et 5 et a uniquement deux inconnues, V6 et V7. Nous supposons que l'outil de résolution résoud notre système et on peut donc marquer V6 et V7 comme connues.

- On résoud ensuite l'équation 3 car elle ne contient plus qu'une seule inconnue. On marque la valeur V4 come connue.

- Il n'y a pas d'équations à une inconnue et on doit donc déterminer un système. Il reste à ce niveau uniquement deux équations, les équations 1 et 2. Quelle que soit l'équation choisie, on aura le même système qui sera composé de ces deux équations. On résoud le système et on marque les variables V2 et V3 comme connues.

- On a déterminé toutes les valeurs initiales, on a terminé.

On peut remarquer que l'algorithme fonctionne toujours aussi bien même s'il y a des systèmes à résoudre.

4.5 Application de la méthode à la régulation adctfree

Le macrobloc de la régulation adctfree étant trop grand, nous avons décidé de le séparer en trois parties afin de pouvoir l'observer. Les trois parties se trouvent aux figures 4.2, 4.3 et 4.4. La première partie correspond à la partie supérieure gauche, la deuxième à la partie supérieure droite et la troisième à la partie inférieure gauche.

On peut voir que c'est une régulation plus importante que celles traitées jusqu'ici et elle a la caractéristique intéressante que les ingénieurs n'arrivent pas à initialiser cette régulation. Il y a 27 équations à 27 variables et ces équations sont "très" non linéaires. En effet, on a, par exemple, des quotients avec des exposants rationnels. Nous n'allons pas, ici, écrire l'ensemble des équations du système car ce n'est pas la chose la plus importante, nous allons plutôt écrire les variables contenues dans chaque équation après avoir précisé les alternatives choisies. Si rien n'est précisé, cela veut dire qu'aucune alternative n'a dû être fixée.

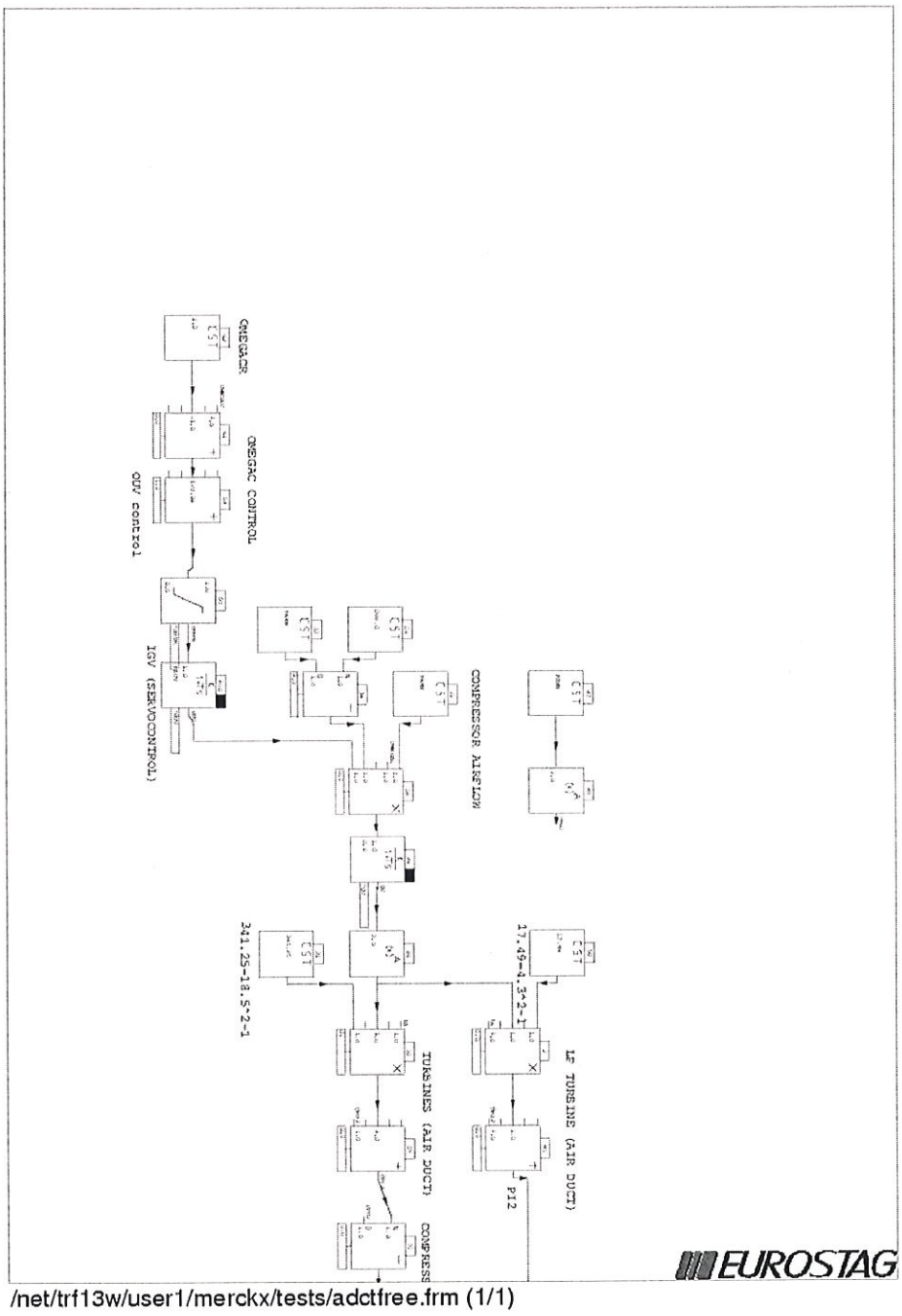

FIG. 4.2 – Première partie du macrobloc de la régulation adctfree

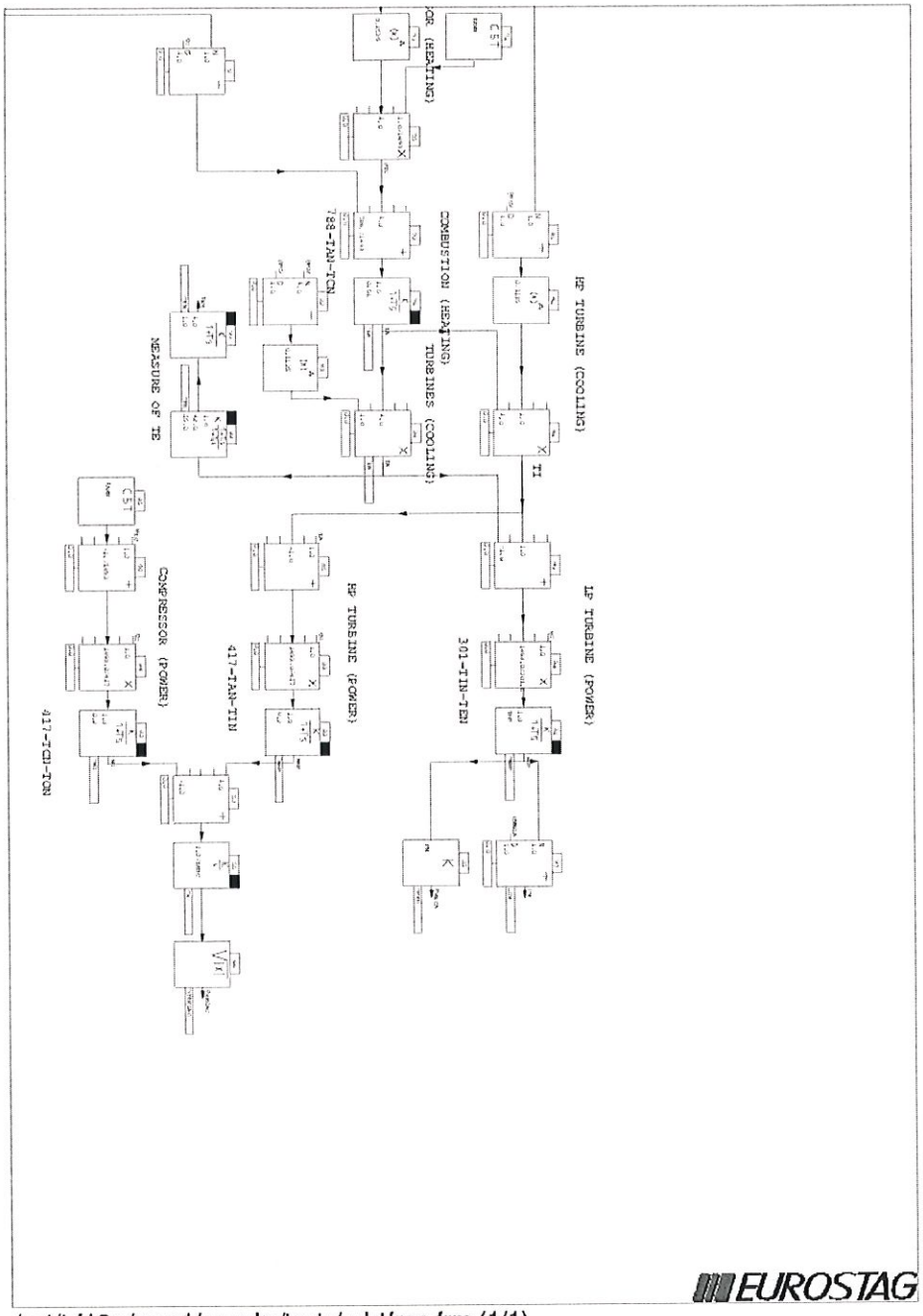

FIG. 4.3 – Deuxième partie du macrobloc de la régulation adctfree

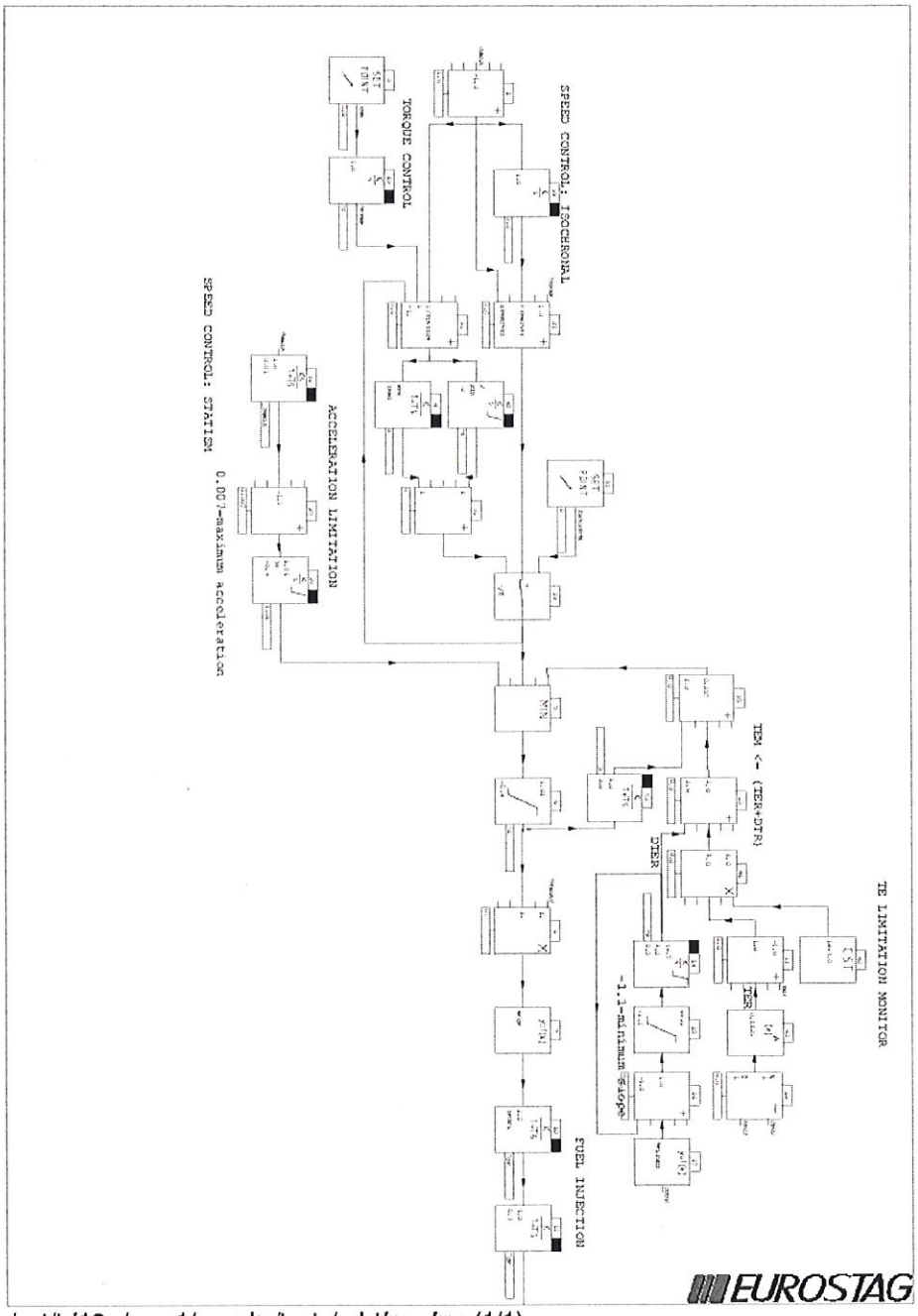

FIG. 4.4 – Troisième partie du macrobloc de la régulation adctfree

1. $f_1(V3) = 0$
2. $f_2(V3) = 0$
3. $f_3(V20) = 0$
4. $f_4(V6, V52) = 0$
5. $f_5(V57, V100) = 0$
6. $f_6(V10, V12) = 0$
7. $f_7(V96, V97) = 0$
8. $f_8(V97) = 0$
9. $f_9(V96, V35) = 0$
10. $f_{10}(V55, V66) = 0$
11. $f_{11}(V26) = 0$
12. $f_{12}(V39, V66, V100) = 0$
13. $f_{13}(V6, V10, V66) = 0$
14. $f_{14}(V88, V93) = 0$
15. $f_{15}(V84, V98) = 0$
16. On a ici la première équation avec alternatives qui correpond au bloc 28. Etant donné que l'entrée du bloc est constante, on sait automatiquement quelle alternative est à choisir, la borne supérieure de l'intervalle. on a donc l'équation $f_{16}(V28) = 0$.
17. Cette équation correspond au bloc 100. On choisira ici l'alternative du milieu de l'intervalle, comme si le bloc limiter n'existe pas, on a alors l'équation $f_{17}(V57, V66) = 0$.
18. Il y a ici un bloc relay et comme la première entrée du bloc peut être déterminée, on peut savoir quelle entrée doit être choisie, la troisième entrée. L'équation devient alors $f_{18}(V4, V13, V40) = 0$. Notons que la deuxième entrée du bloc relay n'est alors jamais utilisée.
19. Cette équation comporte le même bloc relay que la précédente et ce sera donc la troisième entrée qui sera choisie. L'équation devient donc $f_{19}(V4, V13, V40) = 0$.
20. $f_{20}(V12, V39, V79) = 0$
21. $f_{21}(V98, V99) = 0$
22. Il y a ici deux blocs avec alternatives et on supposera que les alternatives choisies pour ces deux blocs sont de passer au milieu de l'intervalle. L'équation devient alors $f_{22}(V14, V57) = 0$.
23. $f_{23}(V39, V79, V93) = 0$
24. $f_{24}(V39, V79, V84) = 0$
25. $f_{25}(V39, V79, V84, V96) = 0$
26. $f_{26}(V39, V79, V88) = 0$

27. Il y a dans cette équation trois blocs avec alternatives. Le premier bloc est le bloc relay et on choisira alors la troisième entrée. Le deuxième bloc est le bloc minimum et on choisira la deuxième entrée. Le dernier bloc est un bloc limiter et nous supposerons que nous passons au travers. L'équation devient alors
$f_{27}(V4, V6, V14, V28, V39, V40, V52, V79) = 0$.

Il faut noter que la valeur initiale $V22$ n'est présente dans aucune équation. Elle fait partie d'une branche inutilisée du bloc relay. L'utilisateur va devoir nous donner la valeur de cette variable.

Nous allons tout d'abord résoudre les équations à une inconnue. On résoud donc les équations 1, 2, 3, 8, 11, 16 et on marque les variables $V3$, $V20$, $V97$, $V26$, $V28$ comme connues. Le système devient alors :

1. $f_4(V6, V52) = 0$
2. $f_5(V57, V100) = 0$
3. $f_6(V10, V12) = 0$
4. $f_7(V96) = 0$
5. $f_9(V96, V35) = 0$
6. $f_{10}(V55, V66) = 0$
7. $f_{12}(V39, V66, V100) = 0$
8. $f_{13}(V6, V10, V66) = 0$
9. $f_{14}(V88, V93) = 0$
10. $f_{15}(V84, V98) = 0$
11. $f_{17}(V57, V66) = 0$
12. $f_{18}(V4, V13, V40) = 0$
13. $f_{19}(V4, V13, V40) = 0$
14. $f_{20}(V12, V39, V79) = 0$
15. $f_{21}(V98, V99) = 0$
16. $f_{22}(V14, V57) = 0$
17. $f_{23}(V39, V79, V93) = 0$
18. $f_{24}(V39, V79, V84) = 0$
19. $f_{25}(V39, V79, V84, V96) = 0$
20. $f_{26}(V39, V79, V88) = 0$
21. $f_{27}(V4, V6, V14, V28, V39, V40, V52, V79) = 0$

On peut maintenant résoudre l'équation 4 du nouveau système. On obtiendra un nouveau système où l'on peut résoudre encore une équation (qui correspond à l'équation 5). On peut donc marquer les variables $V35$ et $V96$ comme connues. On obtient alors le système :

1. $f_4(V6, V52) = 0$
2. $f_5(V57, V100) = 0$
3. $f_6(V10, V12) = 0$
4. $f_{10}(V55, V66) = 0$
5. $f_{12}(V39, V66, V100) = 0$
6. $f_{13}(V6, V10, V66) = 0$
7. $f_{14}(V88, V93) = 0$
8. $f_{15}(V84, V98) = 0$
9. $f_{17}(V57, V66) = 0$
10. $f_{18}(V4, V13, V40) = 0$
11. $f_{19}(V4, V13, V40) = 0$
12. $f_{20}(V12, V39, V79) = 0$
13. $f_{21}(V98, V99) = 0$
14. $f_{22}(V14, V57) = 0$
15. $f_{23}(V39, V79, V93) = 0$
16. $f_{24}(V39, V79, V84) = 0$
17. $f_{25}(V39, V79, V84) = 0$
18. $f_{26}(V39, V79, V88) = 0$
19. $f_{27}(V4, V6, V14, V28, V39, V40, V52, V79) = 0$

On doit alors résoudre un système. Le système déterminé par l'algorithme sera le système comprenant les équations 7, 15, 16, 17 et 18 et impliquant les variables $V39$, $V79$, $V84$, $V88$ et $V93$. On peut le résoudre et marquer ces variables comme connues. Le système devient alors :

1. $f_4(V6, V52) = 0$
2. $f_5(V57, V100) = 0$
3. $f_6(V10, V12) = 0$
4. $f_{10}(V55, V66) = 0$
5. $f_{12}(V66, V100) = 0$
6. $f_{13}(V6, V10, V66) = 0$

7. $f_{15}(V98) = 0$
8. $f_{17}(V57, V66) = 0$
9. $f_{18}(V4, V13, V40) = 0$
10. $f_{19}(V4, V13, V40) = 0$
11. $f_{20}(V12) = 0$
12. $f_{21}(V98, V99) = 0$
13. $f_{22}(V14, V57) = 0$
14. $f_{27}(V4, V6, V14, V28, V40, V52) = 0$

On peut ensuite résoudre les équations 7 et 11 car elles ne contiennent qu'une seule inconnue. On connait donc les variables $V12$ et $V98$. On obtient alors un nouveau système et on peut à nouveau résoudre deux équations qui correspondent aux équations 3 et 12. On connait alors les valeurs des variables $V10$ et $V99$. On obtient alors le système suivant :

1. $f_4(V6, V52) = 0$
2. $f_5(V57, V100) = 0$
3. $f_{10}(V55, V66) = 0$
4. $f_{12}(V66, V100) = 0$
5. $f_{13}(V6, V66) = 0$
6. $f_{17}(V57, V66) = 0$
7. $f_{18}(V4, V13, V40) = 0$
8. $f_{19}(V4, V13, V40) = 0$
9. $f_{22}(V14, V57) = 0$
10. $f_{27}(V4, V6, V14, V28, V40, V52) = 0$

On doit alors résoudre un système d'équations. Le système qu'on obtient un système de 9 équations à 9 inconnues contenant toutes les équations du système ci-dessus sauf la troisième. On résoud le système obtenu. Il ne nous reste plus qu'une variable, $V55$, et une équation. On résoud cette équation.

Chapitre 5

Conclusion

Pour modéliser un réseau électrique, les ingénieurs de Tractebel écrivent les équations sous forme d'un système électrique. Pour simuler le comportement d'un réseau, il est nécessaire de connaître les valeurs initiales des variables d'état qui sont les inconnues de ce système. Les ingénieurs connaissent la valeur initiale de certaines de ces variables par le calcul du load flow et par l'initialisation des machines. Par contre, pour connaître la valeur initiale des autres variables d'état qui correspondent aux régulations des machines, les ingénieurs doivent résoudre un système à la main. La solution de ce système est écrite sous la forme d'un schéma appelé schéma initial.

L'objectif de ce travail était de trouver une méthode permettant de déduire *automatiquement* le schéma initial de n'importe quelle régulation. Nous avons, pour cela, analysé l'ensemble des blocs composant les régulations et déduit une classification de ceux-ci.

Nous avons ensuite déduit une méthode pour les blocs les plus simples et nous avons obtenu de bons résultats avec celle-ci. Nous avons alors tenté de l'enrichir afin que cette méthode puisse traîter les alternatives. Cependant, la méthode que nous avons établie ne nous donnait pas entière satisfaction car il devait y avoir encore une grande intervention de l'utilisateur.

Nous avons alors ajouté des hypothèses au problème afin de pouvoir se ramener à la première méthode quel que soit le macrobloc. Nous avons ensuite testé l'algorithme trouvé sur une régulation déjà analysée par Tractebel et obtenu les mêmes résultats. Puis nous avons tenté d'appliquer l'algorithme à une régulation dont les ingénieurs de Tractebel n'arrivait pas à déduire le schéma initial et notre méthode a pu le trouver.

Cette méthode a été présentée à plusieurs ingénieurs de Tractebel qui initialisent tous les jours plusieurs régulations. La présentation a suscité chez eux de vives réactions. L'idée de l'initialisation automatique est un sujet de conversation et de réflexion récurrent chez eux et ils ont apprécié que le problème soit maintenant clairement établi. A présent, la chose la plus importante pour eux est de pouvoir coder cette méthode afin de voir les résultats qu'elle peut fournir.

Il serait donc intéressant de pouvoir coder la méthode développée dans ce travail afin de tester, pour l'ensemble des régulations de Tractebel, si l'algorithme déduit donne bien de bons résultats. Notons que cela nécessite également le codage d'un outil permettant de résoudre des systèmes non linéaires sous le format utilisé.

Bibliographie

[1] TRACTEBEL-EDF, Eurostag package, *Model Editor User's Manual*, Version 4.2, Octobre 2002.

[2] TRACTEBEL-EDF, Eurostag package, *Tutorial*, Octobre 2002.

[3] TRACTEBEL ENGINEERING, *Chiffre et faits marquants 2003.*

[4] TRACTEBEL ENGINEERING, *http ://www.engineering.tractebel.com.*

Annexe A

Annexe 1 : analyse des blocs

Notons que le bloc Arctan, les blocs logiques et les blocs difficiles n'ont pas été analysés mais ils n'apportent pas grand chose de plus. L'important était d'avoir suffisamment de blocs pour trouver une méthode de résolution et pas l'analyse des blocs.

A.1 Les blocs à valeur connue

Tous les blocs à valeur connue sont à traiter de la même façon :

Description Ils donnent tous une valeur connue.
Valeur Initiale ? Non.
Remplacement Non.
Sens Ils peuvent être uniquement descendus étant donné qu'ils n'ont pas d'entrée.
Résolution Simple

A.2 Le set point

Description Ce bloc donne une valeur initiale. Ce bloc fait partie d'une équation et est obligatoirement une entrée de celle-ci.
Valeur Initiale ? Oui, $y(0)$.
Remplacement Non.
Sens Ce bloc doit être uniquement descendu car il n'a pas d'entrée.
Résolution On doit donc remonter l'équation pour trouver la valeur initiale.

A.3 les blocs algébriques simples avec inverse

- Continuous input delay :

  Description Comme on a fait l'hypothèse que l'état est stationnaire, on a $y(0) = x(0-T) = x(0)$ où T est un décalage de T secondes.

  Valeur Initiale ? Non.

  Remplacement Ce bloc doit être remplacé par un bloc Gain avec un paramètre K=1.

  Sens Ce bloc peut être parcouru dans les deux sens.

  Résolution Simple

- Divider :

  Description On suppose que a_1 et a_2 sont non nuls. On a $y(0) = \frac{a_1 x_1(0)}{a_2 x_2(0)} + b$

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être parcouru dans les deux sens.

  Résolution Selon les cas, on a :

  - Si l'on cherche $y(0)$ (on descend donc le bloc), on a $y(0) = \frac{a_1 x_1(0)}{a_2 x_2(0)} + b$
  - Si on cherche $x_1(0)$ (on remonte le bloc), on a $x_1(0) = \frac{a_2(y(0)-b)x_2(0)}{a_1}$
  - Si on cherche $x_2(0)$ (on remonte le bloc),
    - on a $x_2(0) = \frac{a_1 x_1(0)}{a_2(y(0)-b)}$ si $y(0) \neq b$.
    - Si $y(0) = b$, il faut envisager deux cas :
      - Si $x_1(0) \neq 0$ alors il faut renvoyer une erreur.
      - Si $x_1(0) = 0$ alors impossible de remonter.

- Exponential :

  Description On a $y(0) = A \exp(Bx(0))$.

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être parcouru dans les deux sens.

  Résolution

- On a $y(0) = A\exp(Bx(0))$ quand on descend le bloc.
- Quand on remonte le bloc, on a $x(0) = \frac{1}{B}\ln(\frac{y(0)}{A})$. Il faut alors envisager deux cas :
  - Si $\frac{y(0)}{A} > 0$ alors le ln est bien défini.
  - Sinon, il faut renvoyer une erreur.

- Function :

  Description On a $y(0) = f(x(0))$.

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être uniquement descendu.

  Résolution Il faut juste évaluer la fonction dans le bon intervalle.

- Inverse function :

  Description Ce bloc correspond à la fonction inverse de la fonction du bloc "function".

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être parcouru dans les deux sens étant donné que la fonction est inversible.

  Résolution Si on remonte l'équation, alors le bloc sera à remplacer par le bloc function sinon il restera tel quel.

- Multiplier :

  Description On a, quand on descend le bloc, $y(0) = \prod_i a_i x_i(0) + b$. On peut supposer que les paramètres ne sont pas nuls sinon il n'y a pas lieu de mettre les entrées correspondantes.

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être parcouru dans les deux sens.

  Résolution

- Quand on descend le bloc, $y(0) = \prod_i a_i x_i(0) + b$. On doit donc avoir déterminé toutes les entrées avant de chercher le $y(0)$.

- Quand on remonte le bloc, il y a plusieurs cas à prendre en compte :
  - Si une des entrées est nulle, on ne sait pas remonter le bloc.
  - Par contre, si aucune des entrées n'est nulle, alors on a $x_k(0) = \frac{y(0)-b}{\prod_{i \neq k} a_i x_i(0)} \frac{1}{a_k}$.

- Pulse :

  Description Comme on est en situation stationnaire et que celle-ci dure depuis suffisamment longtemps, on a $y(0) = 0$ quelle que soit l'entrée.

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens On est obligé de descendre ce bloc.

  Résolution Simple.

- Relay :

  Description On peut faire l'hypothèse que l'entrée x_1 est définie. En effet, si elle ne l'était pas, on serait incapable de déterminer la valeur initiale inconnue y figurant.

  Valeur Initiale ? Non.

  Remplacement Non.

  Sens Ce bloc peut être parcouru dans les deux sens.

  Résolution

  - Si l'on descend ce bloc, on évalue $x_1(0)$ et on choisira pour la sortie l'entrée correspondant au signe de x_1.
  - Si on remonte ce bloc, il y a deux cas à envisager. Si on cherche $x_2(0)$ et que $x_1(0)$ est positif, alors on a $x_2(0) = y(0)$. Par contre si $x_1(0)$ est négatif, il est impossible de déterminer $x_2(0)$. La même analyse peut être faite pour x_3.

- Relay with delay :

Description On est dans une situation stationnaire et comme elle dure depuis suffisamment longtemps on peut supposer que $x_1(0)$ est constant depuis suffisamment longtemps.

Sens Ce bloc peut être parcouru dans les deux sens.

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si l'on descend ce bloc, on évalue $x_1(0)$ et on choisira pour la sortie l'entrée correspondant au signe de x_1.
- Si on remonte ce bloc, il y a deux cas à envisager.
  - Si on cherche $x_2(0)$ et que $x_1(0)$ est positif, alors on a $x_2(0) = y(0)$.
  - Par contre si $x_1(0)$ est négatif, il est impossible de déterminer $x_2(0)$. La même analyse peut être faite pour x_3.

- Summer :

Description On a $y(0) = \sum_i a_i x_i(0) + b$.

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc alors on a $y(0) = \sum_i a_i x_i(0) + b$ si toutes les entrées sont connues.
- Si on remonte le bloc, on a $x_k(0) = \frac{y(0) - b - \sum_{i \neq k} a_i x_i(0)}{a_k}$ si toutes les entrées différentes de $x_k(0)$ sont connues et si $a_k \neq 0$. Si $a_k = 0$, on ne peut remonter le bloc pour déduire $x_k(0)$.

A.4 Les blocs différentiels non limités

- Random input Delay :

Description On a $y(0) = x(t \leq 0) = x(0)$.

Valeur Initiale ? Oui, $y(0)$.

Remplacement Ce bloc peut être donc remplacé par un bloc Gain avec K=1.

Sens Dans les deux sens car $y(0)$ pourra être déterminé en remontant une des équation dans laquelle il intervient ou en descendant l'équation associée au bloc.

Résolution Simple

- Derivative lag :

Description On a $y(0) = 0$ et $x(0) = z(0)$. Contrairement aux autres blocs différentiels, ce bloc apporte une valeur initiale et une équation sur cette valeur initiale mais la valeur initiale n'est pas celle du bloc.

Valeur Initiale ? Oui, $z(0)$.

Remplacement Oui, par un bloc Gain avec K=1.

Sens Le bloc doit donc obligatoirement être descendu puisque $z(0)$ n'apparait dans aucune autre équation.

Résolution On doit résoudre l'équation $x(0) = z(0)$.

- Integrator :

Description On sait que $x(0) = 0$ et $y(0)$ est libre.

Valeur Initiale ? Oui, $y(0)$.

Remplacement Le bloc va disparaître du schéma initial. Il va apporter une valeur finale à l'équation qui pourra être remontée.

Sens Le bloc ne peut ni être descendu, ni être remonté puisqu'il va disparaître du schéma initial. Par contre, on va devoir remonter l'équation qui lui est associée.

Résolution On doit donc déterminer $y(0)$ en remontant une des équations dont elle est une des entrées. De plus grâce à l'équation $x(0) = 0$ qu'on pourra remonter, on pourra déterminer une valeur initiale.

- Integrator follower :

Description On peut faire l'hypothèse que l'entrée x_1 est définie. En effet, si elle ne l'était pas, on serait incapable de déterminer la valeur initiale inconnue y figurant.

Valeur Initiale ? Oui, $y(0)$.

Remplacement Cela dépend du signe de $x_1(0)$.
- si $x_1(0) > 0$, alors $x_2(0) = 0$. Donc, on doit remplacer le bloc par un "= 0".
- Si $x_1(0) \leq 0$, alors on a $y(0) = x_3(0)$. On doit donc remplacer le bloc par un bloc Gain avec K=1.

Sens On peut donc remonter ou descendre ce bloc.

Résolution

- Si l'on descend ce bloc, on évalue $x_1(0)$. Si $x_1(0) > 0$, alors $x_2(0) = 0$ et pour trouver $y(0)$, on doit injecter la sortie de ce bloc dans l'équation suivante. En effet, la sortie de ce bloc est aussi l'entrée du bloc suivant qui fait partie lui aussi d'une équation et on pourra ainsi déterminer $y(0)$. Si $x_1(0) \leq 0$, alors on a $y(0) = x_3(0)$.
- Si l'on remonte le bloc, il faut évaluer $x_1(0)$. Si il est négatif alors $x_3(0) = y(0)$ et on ne pourra rien dire à propos de $x_2(0)$. Si $x_1(0) > 0$ alors $x_2(0) = 0$ et on ne pourra rien dire à propos de $x_3(0)$.

- Lead lag filter :

Description On a $y(0) = Kx(0)$.

Valeur Initiale ? Oui, $y(0)$.

Remplacement Ce bloc doit être remplacé par un bloc Gain avec un paramètre K.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution Simple.

- Second order :

Description On a $y(0) = A_0 x(0)$.

Valeur Initiale ? Oui, $y(0)$.

Remplacement On peut donc remplacer ce bloc par un bloc Gain avec comme paramètre A_0.

Sens Ce bloc peut être parcouru dans les deux sens. (l'inconnue peut être déterminée soit en remontant une des équations dont elle est une des entrées soit en descendant l'équation du bloc)

Résolution Simple.

A.5 Les blocs algébriques simples sans inverse et les blocs algébriques avec alternatives

- Abs :

Description $y(0) = |x(0)|$

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut le remonter uniquement si $y \geq 0$.

Résolution

- Si on descend le bloc, on prend pour la sortie la valeur absolue de l'entrée.
- Si on remonte le bloc,
  1. si y(0) est négatif, alors il faut renvoyer une erreur.
  2. Si y(0) est positif, on pourra déterminer l'entrée au signe près : $x(0) = \pm y(0)$.

- Arccos :

Description

$$\begin{array}{rcll} y(0) & = & Arccos(x(0)) & \text{si } -1 < x(0) < 1 \\ & = & Arccos(-1) & \text{si } x(0) \leq -1 \\ & = & Arccos(1) & \text{si } x(0) \geq 1 \end{array}$$

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut le remonter uniquement quand $y(0) \in]0, \pi[$.

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,

1. Si $y(0) \notin [0, \pi]$, on doit renvoyer une erreur.
2. On aura $x(0) = cos(y(0))$ si $y(0) \in]0, \pi[$
3. On ne pourra déterminer l'entrée si $y(0) = 0$ ou si $y(0) = \pi$.

- Arcsin :

Description

$$\begin{array}{rcll} y(0) & = & Arcsin(x(0)) & \text{si } -1 < x(0) < 1 \\ & = & Arcsin(-1) & \text{si } x(0) \leq -1 \\ & = & Arcsin(1) & \text{si } x(0) \geq 1 \end{array}$$

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut le remonter uniquement quand y(0)$\in [\frac{-\pi}{2}, \frac{-\pi}{2}]$

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,
  1. Si $y(0) \notin -\frac{\pi}{2}, \frac{\pi}{2}]$, on doit renvoyer une erreur.
  2. On aura $x(0) = sin(y(0))$ si $y(0) \in] - \frac{\pi}{2}, \frac{\pi}{2}[$.
  3. On ne pourra déterminer l'entrée si $y(0) = -\frac{\pi}{2}$ ou si $y(0) = \frac{\pi}{2}$.

- Cos :

Description $y(0) = cos(x(0))$

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut remonter ce bloc uniquement si $y(0) \in [-1, 1]$

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,

1. Si $y(0) \notin [-1, 1]$, il faut renvoyer une erreur.
2. Sinon, on aura $x(0) = Arccos(y(0)) + k2\pi$, $k \in \mathbb{Z}$.

- Deadband :

Description

$$\begin{array}{rcll} y(0) & = & x(0) - xmin & \text{si } x(0) < xmin \\ & = & 0 & \text{si } xmin \leq x(0) \leq xmax \\ & = & x(0) - xmax & \text{si } x(0) > xmax \end{array}$$

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc, il n'y a pas de problème. On évalue en choisissant la fonction correspondant à l'intervalle auquel appartient l'entrée.
- Si on remonte le bloc, il y a plusieurs cas :
  1. Si $y(0) = 0$, on sait que $x(0) \in [xmin, xmax]$ mais on ne sait pas déteminer plus précisemment sa valeur.
  2. Si $y(0) \neq 0$, il y a deux cas à envisager :
     - Si $y(0) > 0$, alors on a $x(0) = y(0) + xmax$.
     - Si $y(0) < 0$, alors on a $x(0) = y(0) + xmin$.

- Max :

Description $y(0) = max(x_1(0), \ldots, x_n(0))$.

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc, une fois qu'on connait toutes les entrées, on choisit la plus grande entrée.
- Si on remonte le bloc,
  - si on connait toutes les entrées sauf une alors,
    1. Si y(0) est égal à une entrée que l'on connait, on ne peut pas déterminer l'entrée recherchée, on sait juste qu'elle est plus petite que $y(0)$.

2. Si y(0) est différent de toutes les entrées connues alors on sait que l'entrée recherchée est égale à la sortie.

- si toutes les entrées sont connues et que $y(0)$ est différent de chacune d'entre elles, il faut renvoyer une erreur.

- Min :

Description $y(0) = min(x_1(0), \ldots, x_n(0))$.
Valeur Initiale ? Non.
Remplacement Non.
Sens Ce bloc peut être parcouru dans les deux sens.
Résolution

- Si on descend le bloc, une fois qu'on connait toutes les entrées, on choisit la plus petite entrée.
- Si on remonte le bloc,
  - si on connait toutes les entrées sauf une alors,
    1. Si y(0) est égal à une entrée que l'on connait, on ne peut pas déterminer l'entrée recherchée, on sait juste qu'elle est plus grande que $y(0)$.
    2. Si y(0) est différent de toutes les entrées connues alors on sait que l'entrée recherchée est égale à la sortie.
  - si toutes les entrées sont connues et que $y(0)$ est différent de chacune d'entre elles, il faut renvoyer une erreur.

- Power :

Description $y(0) = (x(0))^A$.
Valeur Initiale ? Non.
Remplacement Non.
Sens Ce bloc peut être parcouru dans les deux sens.
Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc, il faut envisager deux cas :

1. Si A est entier,
   - Si A est impair, il n'y a qu'une seule valeur possible pour l'entrée : $x(0) = y(0)^{\frac{1}{A}}$.

- Si A est pair, on peut déterminer l'entrée au signe près : $x(0) = \pm y(0)^{\frac{1}{A}}$. De plus si $y(0) \leq 0$, il faut renvoyer une erreur.

2. Sinon, il n'y a qu'une seule valeur possible pour l'entrée : $x(0) = y(0)^{\frac{1}{A}}$.

- Sin :

Description $y(0) = sin(x(0))$

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut remonter ce bloc uniquement si $y(0) \in [-1, 1]$.

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,
  1. Si $y(0) \notin [-1, 1]$ alors il faut renvoyer une erreur.
  2. Sinon, on aura $x(0) = Arcsin(y(0)) + k2\pi$, $k \in \mathbb{Z}$.

- Square root :

Description $y(0) = \sqrt{|x(0)|}$.

Valeur Initiale ? Non.

Remplacement Non.

Sens

- On peut toujours descendre ce bloc.
- On peut remonter ce bloc uniquement si Si $y(0) \geq 0$

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,
  1. Si $y(0) < 0$, il faut renvoyer une erreur.
  2. Sinon on connait l'entrée au signe près : $x(0) = \pm y(0)^2$.

- Tan :

Description $y(0) = tan(x(0))$.

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc, on a $x(0) = Arctan(y(0)) + k\pi$, $k \in \mathbb{Z}$.

- Variable limiter :

Description

$$\begin{array}{rcll} y(0) & = & x_2(0) & \text{si} \quad x_3(0) \leq x_2(0) \leq x_1(0) \\ & = & x_1(0) & \text{si} \quad x_1(0) < x_2(0) \\ & = & x_3(0) & \text{si} \quad x_2(0) < x_3(0) \end{array}$$

Valeur Initiale ? Non.

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc, on évalue l'entrée.
- Si on remonte le bloc,
  1. Si on connait toutes les entrées sauf une,
     - Si y(0) est égal à une entrée que l'on connait, on ne peut pas déterminer l'entrée recherchée (on connait juste un intervalle auquel cette entrée appartient).
     - Si y(0) est différent de toutes les entrées connues alors on sait que l'entrée recherchée est égale à la sortie.
  2. De plus si toutes les entrées sont connues et que $y(0)$ est différent de chacunes d'entre elles, il faut renvoyer une erreur.

- Zero set counter :

Description Nous avons fait l'hypothèse d'une situation stationnaire au temps initial, il n'y a donc pas de changement de la valeur des entrées. On peut donc faire l'hypothèse que la sortie n'a jamais été incrémentée ni décrémentée.

Valeur Initiale ? Oui, $y(0)$

Remplacement Non.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution

- Si on descend le bloc, il y a deux cas :
  1. Si $x_3(0) < S3$, on a $y(0) = 0$. Si ce résultat ne convient pas avec les équations suivantes, il faudra renvoyer une erreur.
  2. Sinon, la sortie est égale au paramètre à déterminer.
- Si on remonte le bloc, on sait que les deux premières entrées sont libres et on ne pourra les déterminer. Si $y(0) = 0$, on sait que $x_3(0) < S3$. On ne pourra rien dire de plus.

A.6 Les blocs différentiels limités

- Limited integrator :

Description C'est un bloc integrator dont on limite la sortie :

$$\begin{array}{rcll} y(0) & = & Kx & \text{si} \quad ymin < y < ymax \\ & = & & \text{si} \quad y = ymin \text{ et } Kx > 0 \\ & = & & \text{si} \quad y = ymax \text{ et } Kx < 0 \\ & = & 0 & \text{sinon} \end{array}$$

Valeur Initiale ? Oui, $y(0)$.

Remplacement Oui par un bloc Gain avec le paramètre K = 0.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution Il y a plusieurs cas :

- Si le bloc est descendu, il y a trois cas :
  1. Si $x(0) = 0$, alors $y(0)$ est libre. La valeur de $y(0)$ sera à déterminer grâce aux équations suivantes.
  2. Si $Kx(0) < 0$, alors $y(0) = ymin$.
  3. Si $Kx(0) > 0$, alors $y(0) = ymax$.
- Si le bloc est remonté, il y a quatre cas :
  1. Si $y(0) \notin [ymin, ymax]$, alors il faut renvoyer une erreur.
  2. Si $y(0) = ymin$, alors on sait juste que $Kx(0) < 0$.

3. Si $y(0) = ymax$, alors on sait juste que $Kx(0) > 0$.
4. Sinon on a que $x(0) = 0$.

- Limited lead lag filter :

Description C'est un bloc Lead lag filter dont on limite la sortie :

$$\begin{array}{rcll} y(0) & = & Kx(0) & \text{si} \quad ymin < y < ymax \\ & = & & \text{si} \quad y = ymin \text{ et } Kx \geq ymin \\ & = & & \text{si} \quad y = ymax \text{ et } Kx \leq ymax \\ & = & 0 & \text{sinon} \end{array}$$

Valeur Initiale ? Oui, $y(0)$.

Remplacement Oui, par un bloc Gain avec un paramètre K suivi d'un bloc limiter de paramètre Ymin et Ymax.

Sens Ce bloc peut être parcouru dans les deux sens.

Résolution Il y a plusieurs cas à envisager :

- Si on descend le bloc, il y a trois cas :
  1. Si $Kx(0) < ymin$, alors $y(0) = ymin$
  2. Si $Kx(0) > ymax$, alors $y(0) = ymax$
  3. Sinon $y(0) = Kx(0)$
- Si on remonte le bloc,
  1. Si $y(0) \notin [ymin, ymax]$, alors il faut renvoyer une erreur.
  2. Si $y(0) = ymin$, alors on sait juste que $Kx(0) < 0$.
  3. Si $y(0) = ymax$, alors on sait juste que $Kx(0) > 0$.
  4. Sinon $y(0) = Kx(0)$