

République Algérienne Démocratique et Populaire

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERY, Tizi-Ouzou**



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme

*MASTER ACADEMIQUE EN AUTOMATIQUE
OPTION : COMMANDE DES SYSTEMES*

Thème

Commande optimale par l'approche indirecte basée sur la
méthode de décomposition d'adomian

Proposé par : M.A.MAIDI
Dirigé par : M.A.AKKOUCHE

Présenté par :
M^{lle} BRAHAMI Cherifa

Soutenu le : / /2013

Promotion 2013

Remerciement

Je désire d'abord et avant tout remercier le bon dieu qui m'a donné le courage, l'aptitude et le sérieux de mener ce travail.

Ensuite, Il m'est agréable d'exprimer ma profonde gratitude à mon promoteur A.MAIDI qui m'a aidé et soutenu tout au long de mon travail, et qui n'a jamais manqué de m'orienter et de me conseiller. Qu'il trouve ici l'expression de mon respect et de ma profonde reconnaissance.

Mes vifs remerciements à mon Co-promoteur, A.AKKOUCHE qui m'a aidé à l'élaboration de mon mémoire. Avec vous j'ai beaucoup appris.

Mes remerciements les plus sincères vont à Mesdames et Messieurs les membres du jury, qui m'ont fait l'honneur d'accepter de juger ce modeste travail. Pour cela, ainsi que pour leurs commentaires sur le mémoire, je leur exprime ma profonde gratitude.

Je suis aussi reconnaissante à tous les enseignants qui m'ont soutenu tout au long de mes études. Qu'ils trouvent ici l'expression de mes sentiments les plus respectueux.

Enfin je dédie mon travail à ma famille qui m'a toujours soutenu et encouragé et à qui j'exprime toute ma gratitude. Tous les sacrifices que vous avez consentit pour cette thèse me vont droit au cœur !

Sommaire

Introduction générale	1
Chapitre I	Commande optimale
I.1 Introduction.....	2
I.2 Commande optimale	2
I.3 formulation mathématique d'un problème de commande optimale	3
I.3.1 Mise en équation.....	3
I.3.2 Conditions terminales	3
I.3.3 Critère de performance	4
I.3.3.1 Commande en temps minimal.....	4
I.3.3.2 La poursuite	5
I.3.3.3 La régulation.....	5
I.3.3.4 La commande à énergie minimale.....	5
I.4 Exemple illustratif.....	7
I.5 Synthèse d'une loi de commande optimale	9
I.6 Conclusion	10
Chapitre II	La méthode de décomposition d'Adomian
II.1 Introduction	11
II.2 Description de la méthode de décomposition d'Adomian.....	11
II.2.1 Exemple1.....	15
II.2.2 Exemple2.....	19
II.3 Conclusion	21
Chapitre III	L'optimisation
III.1 Introduction	22
III.2 Problèmes d'optimisations	22
III.2.1 Terminologie	22
III.2.2 Les problèmes d'optimisation sans contraintes.....	22
III.3 Concepts de base	23
III.3.1 Gradient.....	23
III.3.2 Hessien	23
III.3.3 La convexité	26
III.3.3.1 Ensembles convexes	26
III.3.3.2 Fonctions convexes	27

III.3.3.2.1 Fonctions convexes différentiables	27
III.3.3.2.2 Condition nécessaire et suffisante d'optimalité globale.....	27
III.4 Méthode de Newton	28
III.4.1 Introduction	28
III.4.2 Le pas de Newton.....	28
III.4.3 La longueur du pas	29
III.4.3.1 Condition d' Armijo.....	29
III.4.3.2 Condition de Wolfe.....	30
III.4.3.3 Algorithme de recherche linéaire	30
III.4.4 Algorithme de Newton.....	31
III.5 Conclusion.....	33
ChapitreIV	
Calcul de la loi de commande par l'Approche Indirecte	
IV.1 Introduction.....	34
IV.2 Principe de l'Approche	34
IV.2.1 Procédure	35
IV.3 Application.....	36
IV.4 Conclusion	42
Conclusion générale	43

Un des traits plus typiques de notre époque est l'intérêt croissant aux problèmes de commande, de contrôle et de gestion lié à la nécessité d'une gestion efficace des ressources naturelles et humaines, des moyens matériels et techniques d'où la naissance de la commande optimale qui a fait l'objet d'un très large développement méthodologique.

La commande optimale consiste à synthétiser des lois de commandes optimales qui minimisent un critère exprimant les objectifs désirés à optimiser, tout en respectant un certain nombre de contraintes et de conditions terminales [1,6 ,7].

Toute recherche de commande optimale nécessite la manipulation d'expressions mathématiques et en particulier celles caractérisant l'évolution du processus, c'est-à-dire de son modèle. Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés.

L'optimisation est un sujet très ancien qui connaît un nouvel essor depuis l'apparition des ordinateurs et dont les méthodes s'appliquent dans de très nombreux domaines, économie, gestion, automatique, robotique, conception optimale, etc [3, 4,9,11].

L'objectif de notre travail est d'utiliser l'approche indirecte pour la détermination de la commande optimale d'un système dynamique. L'idée consiste à transformer le problème de commande optimale en un problème d'optimisation en utilisant la méthode de décomposition d'Adomian[2, 8]. Une fois le problème d'optimisation est obtenu, on applique une méthode d'optimisation classique pour déterminer les paramètres de la loi de commande optimale [10].

Ce mémoire comporte quatre chapitres :

Dans le premier chapitre, on présente des généralités sur la commande optimale et la démarche à suivre pour formuler un problème de commande optimale[1].

Le second chapitre expose la méthode de décomposition d'Adomian [2, 8] qui permet de résoudre des équations différentielles simplement en utilisant un schéma itératif.

Le troisième chapitre donne quelques notions sur l'optimisation et la méthode d'optimisation numérique de Newton [3, 4,9,11].

Le dernier chapitre est consacré à la résolution du problème de commande optimale en utilisant la combinaison de la méthode d'Adomian et la méthode d'optimisation de Newton.

La fin du mémoire est réservée à une conclusion sur l'ensemble du travail.

I.1 Introduction :

La recherche d'une commande permettant d'atteindre un certain nombre d'objectifs tout en minimisant ou maximisant un critère donné, constitue le problème fondamental de la commande optimale.

Le problème général de la détermination d'une commande optimale d'un processus peut se résumer comme suit :

Un processus étant donné et défini par son modèle, trouver une commande qui permet à la fois :

1. De vérifier des conditions initiales et finales données
2. D'optimiser un critère choisi

I.2 Commande optimale : [1]

La commande optimale consiste à déterminer une commande permet de transférer l'état du système $x(t)$ de l'état initial $x(t_0)$ à l'état final $x(t_f)$ tout en optimisant un critère de performance.

Le temps de transfert $t_f - t_0 = T$ est appelé « Horizon » de commande. Ce dernier peut être fini ou infini

L'existence d'une commande satisfaisant un critère de performance choisi, suppose que le système soit commandable, c'est une hypothèse qui sera faite implicitement de façon systématique le long de notre travail.

I.3 formulation mathématique d'un problème de commande optimale :

I.3.1 Mise en équation :

Un système dynamique, ou processus, est caractérisé par trois ensembles de variables :

1. Les variables de sorties, en général directement accessible et regroupées dans un vecteur y de dimension « r ».
2. Les variables de commandes, regroupées dans un vecteur u de dimension « p » et dont le choix permet d'agir sur l'évolution du processus.
3. Les variables internes caractérisant l'état du processus à un instant donné regroupées dans un vecteur d'état $x(t)$ de dimension « l ». l'instant courant est noté $t > 0$.

Dans une modélisation de l'évolution du processus, ces diverses variables sont liées par une équation d'état le plus souvent explicitée sous la forme :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t), u(t)) \end{cases} \quad (I.1)$$

Où $x(t) \in \mathfrak{R}^l$, $u \in \mathfrak{R}^p$, $y \in \mathfrak{R}^r$ et $t \in \mathfrak{R}^+$

I.3.2 Conditions terminales :

Les conditions terminales caractérisant à la fois l'état initial, c'est-à-dire l'instant où on commence à agir sur le processus, et l'état final, après action de la commande.

L'instant initial est noté « t_0 » et l'état initial qui est toujours connu est noté $x_0 = x(t_0)$. De même, l'instant final est noté « t_f » et l'état final qui est, soit imposé (connu) ou libre (inconnu) $x_f = x(t_f)$.

Lorsque l'état final est imposé, soit il est donné directement, soit il sera imposé par une autre condition.

I.3.3 Critère de performance :

Ce critère doit être choisi selon les objectifs désirés (Poursuite, Régulation, Temps minimal, Energie minimale, Consommation minimale,...), il est donné sous la forme suivante :

$$J(u(t)) = \underbrace{\Psi(x(t_f), t_f)}_{\text{Partie terminale}} + \underbrace{\int_{t_0}^{t_f} G(x(t), u(t), t) dt}_{\text{Partie intégrale}} \quad (\text{I.2})$$

La partie terminale exprime l'objectif à optimiser à l'instant final par contre la partie intégrale exprime les objectifs à optimiser sur l'horizon de commande.

Selon la forme de critère, on distingue trois types de problème de commande optimale :

1. Problème de Mayer (Partie terminale).
2. Problème de Lagrange (Partie intégrale).
3. Problème de Bolza (Partie terminale et intégrale).

Les critères de performances les plus utilisés sont :

I.3.3.1 Commande en temps minimal :

Le but est de conduire le système d'un état initial x_0 à un état final x_f en minimisant le temps. Le critère utilisé s'écrit :

$$\min T = \min_{t_f} \int_{t_0}^{t_f} 1 dt \quad (\text{I.3})$$

Par identification, on trouve : $\Psi = 0$ et $G = 1$.

I.3.3.2 Poursuite :

Il s'agit de maintenir l'état $x(t)$ du système très proche de l'état désiré $x^d(t)$ dans l'intervalle de temps $[t_0, t_f]$. Le critère correspondant est :

$$J(u(t)) = \int_{t_0}^{t_f} [x^d(t) - x(t)]^T G [x^d(t) - x(t)] dt \quad (\text{I.4})$$

Avec $G = G^T$, $G > 0$

I.3.3.3 Régulation :

La régulation c'est un cas particulier de la poursuite, dans ce cas on a $x^d(t) = 0$ avec $t \in [t_0, t_f]$. Le critère est donné comme suit :

$$J(u(t)) = \int_{t_0}^{t_f} [x(t)]^T G [x(t)] dt \quad (\text{I.5})$$

Avec $G = G^T$, $G > 0$

I.3.3.4 Commande à énergie minimale :

Elle consiste à conduire le système d'un état initial x_0 à l'état final x_f en minimisant l'effort de la commande. Le critère utilisé se formule de la façon suivante :

$$J(u(t)) = \int_{t_0}^{t_f} u^T(t).R.u(t) dt \quad (\text{I.6})$$

Avec $R = R^T$, $R > 0$

Ces critères peuvent être combinés si on a plusieurs objectifs,

Par exemple: Le critère de performance choisi pour une commande en temps minimal et énergie minimale est donné comme suit:

$$J(u(t)) = \underbrace{\int_{t_0}^{t_f} 1 dt}_{\text{Temps minimal}} + \underbrace{\int_{t_0}^{t_f} u^2(t) dt}_{\text{Energie minimale}} \quad (\text{I.7})$$

En résumé, On définit un problème général de commande optimale, sous la forme de Bolza par :

$$\left\{ \begin{array}{l} \min_{u(t)} J(u(t)) = \underbrace{\Psi(x(t_f), t_f)}_{\text{Partie terminale}} + \underbrace{\int_{t_0}^{t_f} G(x(t), u(t), t) dt}_{\text{Partie intégrale}} \\ \text{Sous des contraintes} \\ \dot{x}(t) = f(x(t), u(t)) \end{array} \right.$$

Avec les conditions aux limites :

$$x(t_0) = x_0,$$

$$x(t_f) = x_f$$

Où $x(t) \in \mathfrak{R}^l$ est le vecteur d'état, $u(t) \in \mathfrak{R}^p$ est le vecteur de contrôle. t_0 et t_f sont respectivement le temps initial et le temps final. $G: \mathfrak{R}^l \times \mathfrak{R}^p \times \mathfrak{R}$ est la fonctionnelle, et Ψ est une fonction scalaire. x_0 et x_f sont respectivement l'état initial et l'état final du système, avec x_f peut être fixe ou libre.

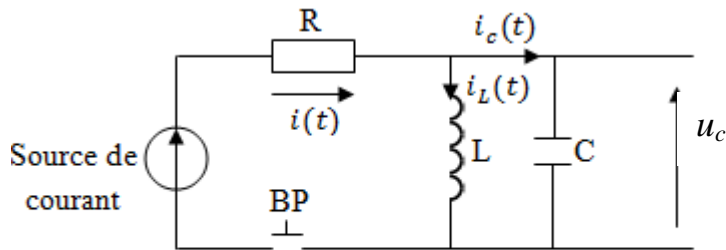
I.4 Exemple illustratif :

Figure (I.1) : circuit électrique

Pour le circuit de la figure (I.1), on désire déterminer la commande optimale $u^*(t)$ permettant de transférer le système d'un état initial à un état final.

Un instant final « t_f » caractérisé par une tension aux bornes de la capacité de « 1v » et un courant « 0A » dans l'inductance « L », tout en minimisant la perte d'énergie par effet Joule dans la résistance R .

A l'instant « $t = 0$ », on ferme l'interrupteur « BP ».

1. Le modèle :

On pose :

$$x_1(t) = i_L(t), \quad x_2(t) = u_c(t), \quad u(t) = i(t)$$

D'après le schéma de la figure (1) et la loi des mailles

$$-\frac{di_L(t)}{dt} + u_c(t) = 0$$

D'où

$$\dot{x}_1(t) = \frac{1}{L} x_2(t)$$

D'après la loi des Nœuds

$$i(t) = i_L(t) + i_c(t)$$

D'autre part

$$U_c(t) = \frac{1}{C} \int i_c(t) dt$$

$$i_c(t) = C \frac{du_c(t)}{dt}$$

D'où

$$i(t) = i_L(t) + C \frac{du_c(t)}{dt}$$

$$\frac{du_c(t)}{dt} = \frac{1}{C} [i(t) - i_L(t)]$$

$$\dot{x}_2(t) = -\frac{1}{C} x_1(t) + \frac{1}{C} u(t)$$

Finalement le modèle obtenu s'écrit comme suit :

$$\begin{cases} \dot{x}_1(t) = \frac{1}{L} x_2(t) \\ \dot{x}_2(t) = -\frac{1}{C} x_1(t) + \frac{1}{C} u(t) \end{cases}$$

3. Les conditions terminales:

2.1 Conditions initiales :

$$\text{A } \ll t = 0 \gg x(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{2.2- Conditions finales : A } \ll t = t_f \gg x(t_f) = \begin{bmatrix} x_1(t_f) \\ x_2(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

3. Le critère à optimiser :

L'objectif :

- Energie minimale.

$$\min j(u(t)) = \int_0^{t_f} i(t)u_R(t)dt$$

$$\min j(u(t)) = \int_0^{t_f} Ri^2(t)dt = \int_0^{t_f} u^T(t)Ru(t)dt$$

D'où le problème s'écrit comme suit :

$$\left\{ \begin{array}{l} \min j(u(t)) = \int_0^{t_f} u^T(t)Ru(t)dt \\ \text{S. à} \\ \dot{x}_1(t) = \frac{1}{L}x_2(t) \\ \dot{x}_2(t) = -\frac{1}{C}x_1(t) + \frac{1}{C}u(t) \\ x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ x(t_f) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{array} \right.$$

I.5 Synthèse d'une loi de commande optimale :

La synthèse d'une loi de commande optimale passe par les étapes suivantes :

1. Modélisation du procédé à commander.
3. Choix étudié avec soin du critère de performance à minimiser.
4. Résolution du problème pour déterminer la loi de commande optimale $\dot{u}(t)$.
5. Implémentation de la loi de commande.

I.6 Conclusion :

L'objectif de ce chapitre était de mettre en évidence les éléments nécessaires et la démarche à suivre pour la formulation d'un problème de commande optimale. Une fois le modèle est obtenu, l'étape suivante consiste à déterminer la loi de commande.

Dans ce mémoire, on suppose que la commande soit donnée sous forme d'une série de fonctions pondérées par des paramètres à déterminer de façon à minimiser le critère. Ce qui permet de ramener le problème de commande optimale à un problème d'optimisation classique. Pour ce faire, on utilise la méthode de décomposition d'Adomian qui est décrite dans le chapitre suivant.

II.1 Introduction :

Le présent chapitre est consacré à la présentation d'une méthode qui nous permet de résoudre les équations différentielles qui est la méthode de décomposition d'Adomian.

La méthode de décomposition d'Adomian permet de résoudre des problèmes fonctionnels de différents types : équations algébriques, intégrales, aux dérivées partielles (EDP), différentielles,....

II.2 Description de la méthode de décomposition d'Adomian :[2,8]

L'application de la méthode de décomposition d'Adomian pour la résolution des systèmes d'équations peut être résumée de la façon suivante :

Soit une équation différentielle donnée sous forme :

$$x^{(q)}(t) + x^{(m)}(t) = g(t) \quad (\text{II. 1})$$

Avec $q > m$

Tel que :

$x(t)$: est la solution de l'équation donnée par la formule (II.1)

q : est la q^e dérivée de $x(t)$.

m : est la m^e dérivée de $x(t)$.

$g(t)$: est le seconde membre.

La méthode de décomposition d'Adomian nous permet d'introduire des opérateurs linéaires « L » et « R » afin d'écrire l'équation (II.1) sous la forme :

$$Lx(t) + Rx(t) = g(t) \quad (\text{II. 2}).$$

Tel que :

$$L = \frac{d^q}{dt^q} ; R = \frac{d^m}{dt^m} \quad (\text{II. 3})$$

Avec :

L : correspond au degré le plus élevé.

R : correspond au degré le moins élevé.

Supposons L est inversible et son inverse :

$$L^{-1} = \underbrace{\int_0^t \int_0^t \dots \int_0^t (\cdot) dt dt \dots dt}_{q \text{ intégrale}} \quad (\text{II. 4})$$

Si on introduit L^{-1} dans l'équation (II.2) on aura :

$$L^{-1}Lx(t) + L^{-1}Rx(t) = L^{-1}g(t) \quad (\text{II. 5})$$

Alors, on obtient :

$$\begin{aligned} x(t) - h(t) &= L^{-1}g(t) - L^{-1}R(x(t)) \\ x(t) &= L^{-1}g(t) + h(t) - L^{-1}R(x(t)) \end{aligned} \quad (\text{II. 6})$$

avec $h(t)$ est la fonction obtenue en considérant les conditions initiales du problème.

On pose :

$$L^{-1}g(t) + h(t) = E(t) \quad (\text{II. 7})$$

Alors l'équation (II.6), devient :

$$x(t) = E(t) - L^{-1}R(x(t)) \quad (\text{II. 8})$$

Selon la méthode de décomposition d'Adomian, la solution $x(t)$ s'écrit sous forme d'une série infinie donnée comme suit :

$$x(t) = \sum_{n=0}^{\infty} x_n(t), \quad (\text{II. 9})$$

Où les termes $x_n(t)$, $n \geq 0$ sont déterminés itérativement.

En remplaçant l'équation (II.9) dans l'équation (II.8), on aura :

$$\sum_{n=0}^{\infty} x_n(t) = E(t) - L^{-1} \left(R \sum_{n=0}^{\infty} x_n(t) \right) \quad (\text{II. 10})$$

D'où les termes $x_n(t)$ peuvent se déduire comme suit :

$$x_0(t) = E(t),$$

$$x_1(t) = -L^{-1} \left(R \left(x_0(t) \right) \right),$$

$$x_2(t) = -L^{-1} \left(R \left(x_1(t) \right) \right),$$

⋮

$$x_{k+1}(t) = -L^{-1} \left(R \left(x_k(t) \right) \right),$$

Ce qui entraîne d'obtenir le schéma itératif suivant

$$\begin{cases} x_0(t) = E(t) \\ x_{k+1}(t) = -L^{-1} \left(R \left(x_k(t) \right) \right) \end{cases} \quad (\text{II. 11})$$

Une fois les termes $x_k(t)$ sont déterminés, la solution $\phi_n(t)$ peut être donnée sous forme d'une série infinie

$$\phi_n(t) = \sum_{k=0}^n x_k(t) \quad (\text{II. 12})$$

Par conséquent, la solution $x(t)$ est donnée par :

$$x(t) = \lim_{n \rightarrow \infty} \phi_n(t) \quad (\text{II. 13})$$

La méthode de décomposition d'Adomian s'adapte aussi bien aux problèmes linéaires qu'aux problèmes non linéaires. Il suffit qu'on puisse écrire l'équation sous la forme :

$$x - Nx = g \quad (\text{II. 14})$$

Qui est appelée forme canonique d'Adomian

Où N est un opérateur différentiel non linéaire et g une fonction connue.

La méthode de décomposition d'Adomian consiste à rechercher la solution x sous forme d'une série infinie:

$$\sum_{n=0}^{+\infty} x_n \quad (\text{II. 15})$$

Et à décomposer le terme non linéaire Nx sous forme d'une série infinie :

$$Nx = \sum_{n=0}^{+\infty} A_n \quad (\text{II. 16})$$

Les termes A_n sont appelés polynômes d'Adomian et sont obtenus par la relation suivante :

$$A_n = \frac{1}{n!} \frac{d^n}{dt^n} \left[N \left(\sum_{i=0}^n \lambda^i x_i \right) \right]_{\lambda=0} \quad (\text{II. 17})$$

Où λ est un paramètre réel introduit par convenance.

En remplaçant les relations (II.15) et (II.16) dans (II.14), on obtient :

$$\sum_{n=0}^{+\infty} x_n = g + \sum_{n=0}^{+\infty} A_n \quad (\text{II. 18})$$

Ce qui entraîne par identification :

$$\begin{cases} x_0 = g(t) \\ x_1 = A_1 \\ \vdots \\ x_{k+1} = A_k \end{cases} \quad (\text{II.19})$$

Il est noté que cette identification n'est pas unique mais c'est la seule qui permet de définir explicitement les x_n . La relation (II.19) permet de calculer tous les termes de la série.

En résumé, après la détermination des A_n , une sommation donne la solution sous forme :

$$\Phi_n(t) = \sum_{k=0}^n x_k(t) \quad (\text{II.20})$$

Par conséquent, la solution $x(t)$ est donnée par :

$$x(t) = \lim_{n \rightarrow \infty} \Phi_n(t) \quad (\text{II.21})$$

II.2.1 Exemple1

Soit le système décrit par l'équation linéaire suivante

$$\ddot{x}(t) - x(t) = 0$$

$$\text{Telque } \dot{x}(0) = A$$

Avec une condition initiale: $x(0) = 1$

En appliquant la méthode d'Adomian décrite précédemment, on aura

$$Lx(t) - x(t) = 0 \quad (\text{II.22})$$

$$\text{Tel que } L = \frac{d^2}{dt^2} \text{ et } L^{-1} = \int_0^t \int_0^t (\cdot) dt dt$$

Nous introduisons l'opérateur L^{-1} dans l'équation(II.22), on obtient

$$L^{-1}Lx(t) = L^{-1}x(t)$$

On calcule le terme $L^{-1}Lx(t)$:

$$L^{-1}Lx(t) = \int_0^t \int_0^t \ddot{x}(t) dt dt$$

$$L^{-1}Lx(t) = \int_0^t \dot{x}(t)|_0^t dt$$

$$L^{-1}Lx(t) = \int_0^t (\dot{x}(t) - \dot{x}(0)) dt$$

$$L^{-1}Lx(t) = \int_0^t (\dot{x}(t) - A) dt$$

$$L^{-1}Lx(t) = (x(t) - At)|_0^t$$

$$L^{-1}Lx(t) = x(t) - At - x(0)$$

D'où :

$$L^{-1}Lx(t) = -1 - At + x(t)$$

Donc

$$x(t) = 1 + At + L^{-1}x(t)$$

Avec

$$x(t) = \sum_{n=0}^{\infty} x_n(t)$$

On aura :

$$\sum_{n=0}^{\infty} x_n(t) = 1 + At + L^{-1} \left[\sum_{n=0}^{\infty} x_n(t) \right]$$

Et le schéma itératif est donné sous forme

$$\begin{cases} x_0(t) = 1 + At \\ x_{k+1}(t) = L^{-1}x_k(t) \end{cases}$$

Pour $k = 0$

$$x_1(t) = L^{-1}x_0(t)$$

$$x_1(t) = \int_0^t \int_0^t x_0(t) dt dt$$

$$x_1(t) = \int_0^t \int_0^t (1 + At) dt dt$$

$$x_1(t) = \int_0^t \left[\left(t + \frac{1}{2} At^2 \right) \Big|_0^t \right] dt$$

$$x_1(t) = \left(\frac{1}{2} t^2 + \frac{1}{6} At^3 \right) \Big|_0^t$$

$$x_1(t) = \frac{1}{2} t^2 + \frac{1}{6} At^3 = \frac{1}{2!} t^2 + \frac{1}{3!} At^3$$

Pour $k = 1$

$$x_2(t) = L^{-1}x_1(t)$$

$$x_2(t) = \int_0^t \int_0^t x_1(t) dt dt$$

$$x_2(t) = \int_0^t \int_0^t \left(\frac{1}{2} t^2 + \frac{1}{6} At^3 \right) dt dt$$

$$x_2(t) = \int_0^t \left(\frac{1}{6} t^3 + \frac{1}{24} At^4 \right) dt$$

$$x_2(t) = \frac{1}{24} t^4 + \frac{1}{120} At^5 = \frac{1}{4!} t^4 + \frac{1}{5!} At^5$$

Pour $k = 2$

$$x_3(t) = L^{-1}x_2(t)$$

$$x_3(t) = \int_0^t \int_0^t \left(\frac{1}{24} t^4 + \frac{1}{120} At^5 \right) dt dt$$

$$x_3(t) = \int_0^t \left(\frac{1}{120} t^5 + \frac{1}{720} At^6 \right) dt$$

$$x_3(t) = \frac{1}{720}t^6 + \frac{1}{5040}At^7 = \frac{1}{6!}t^6 + \frac{1}{7!}At^7$$

⋮

Pour $k = n - 1$

$$x_n(t) = \frac{t^{2n}}{2n!} + A \frac{t^{(2n+1)}}{(2n+1)!}$$

$$\Phi_n(t) = \sum_{n=0}^{+\infty} x_n(t)$$

$$\begin{aligned} \Phi_n(t) = & (1 + At) + \left(\frac{1}{2!}t^2 + \frac{1}{3!}At^3\right) + \left(\frac{1}{4!}t^4 + \frac{1}{5!}At^5\right) + \left(\frac{1}{6!}t^6 + \frac{1}{7!}At^7\right) + \dots \\ & + \left(\frac{t^{2n}}{2n!} + A \frac{t^{(2n+1)}}{(2n+1)!}\right) \end{aligned}$$

$$\begin{aligned} \Phi_n(t) = & \left(1 + \frac{1}{2!}t^2 + \frac{1}{4!}t^4 + \frac{1}{6!}t^6 + \dots + \frac{t^{2n}}{2n!}\right) \\ & + A \left(t + \frac{1}{3!}t^3 + \frac{1}{5!}t^5 + \frac{1}{7!}t^7 + \dots + \frac{t^{(2n+1)}}{(2n+1)!}\right) \end{aligned}$$

D'où

$$x(t) = ch(t) + Ash(t)$$

Pour calculer A , on donne par exemple les conditions terminales suivantes :

Pour $t \in [0, 1]$

$$\begin{cases} x(0) = 1 \\ x(1) = 0 \end{cases}$$

$$x(1) = ch(1) + Ash(1)$$

Ce qui implique

$$A = -\frac{ch(1)}{sh(1)} = -\frac{e^2 + 1}{e^2 - 1}$$

II.2.2 Exemple2

Soit l'équation non linéaire suivante :

$$\dot{x}(t) + x^2(t) = 0 \text{ Avec } x(0) = 1$$

$$Lx(t) + N(x(t)) = 0$$

$$L^{-1}Lx(t) = -L^{-1}[N(x(t))]$$

$$\int_0^t \dot{x}(t)dt = -L^{-1}[N(x(t))]$$

$$x(t) = -1 - L^{-1}[N(x(t))]$$

Tel que

$$\begin{cases} N(x(t)) = \sum_{n=0}^{\infty} A_n \\ x(t) = \sum_{n=0}^{\infty} x_n(t) \end{cases}$$

Ce qui donne l'équation suivante

$$\sum_{n=0}^{\infty} x_n(t) = -1 - L^{-1} \left[\sum_{n=0}^{\infty} A_n \right]$$

Par identification, on aura le schéma itératif suivant

$$\begin{cases} x_0 = -1 \\ x_1 = -L^{-1}(A_0) \\ \vdots \\ x_{k+1} = -L^{-1}(A_k) \end{cases}$$

On calcule les polynômes A_n

Pour $k = 0$

$$A_0 = [N(\lambda^0 x_0)]_{\lambda=0}$$

$$A_0 = x_0^2$$

Pour $k = 1$

$$A_1 = \frac{1}{1!} \left[\frac{d}{d\lambda} (N\{x_0 + \lambda x_1\}) \right]_{\lambda=0}$$

$$A_1 = \left[\frac{d}{d\lambda} (x_0^2 + 2\lambda x_0 x_1 + \lambda^2 x_1^2) \right]_{\lambda=0}$$

$$A_1 = 2x_0 x_1$$

De même procédure, on aura

$$A_2 = 2x_0 x_2 + x_1^2$$

$$A_3 = 2x_1 x_2 + 2x_0 x_3$$

⋮

D'après le schéma itératif, on a

Pour $k = 0$

$$x_1 = -L^{-1}(A_0) = -L^{-1}(x_0^2)$$

$$x_1 = -L^{-1}(1)$$

$$x_1 = -t$$

Pour $k = 1$

$$x_2 = -L^{-1}(A_1) = -L^{-1}(2x_0 x_1)$$

$$x_2 = -\int_0^t 2t \, dt$$

$$x_2 = -t^2$$

Pour $k = 2$

$$x_3 = -L^{-1}(A_2) = -L^{-1}(2x_0 x_2 + x_1^2)$$

$$x_3 = -\int_0^t 3t^2 \, dt$$

$$x_3 = -t^3$$

Pour $k = 3$

$$x_4 = -L^{-1}(A_3)$$

$$x_4 = -\int_0^t (2x_1x_2 + 2x_0x_3)dt$$

$$x_4 = -t^4$$

⋮

D'où :

$$\Phi_n(t) = \sum_{n=0}^{+\infty} x_n(t) = -(1 + t + t^2 + t^3 + t^4 + \dots)$$

$$x(t) = \lim_{n \rightarrow \infty} \Phi_n(t) = -\frac{1}{1-t}$$

II.3 Conclusion :

Dans ce chapitre on a présenté la méthode de décomposition d'Adomian qui permet de résoudre des équations différentielles linéaires et non linéaires. Cette méthode permet de transformer ce système à un problème d'optimisation. Le chapitre suivant est consacré aux méthodes d'optimisation.

III.1 Introduction

Le concept de la convexité a une grande importance dans l'étude des problèmes d'optimisation. Dans ce chapitre, nous allons présenter quelques notions relatives à l'optimisation et quelques méthodes de détermination de l'optimum d'une fonction.

III.2 Problèmes d'optimisation[3]

III.2.1 Terminologie

On définit un problème d'optimisation par :

$$\begin{aligned} & \text{Minimiser } f(x) \\ \text{S.C. } & \quad g_i(x) \leq 0, \quad i = 1, \dots, \mu \\ & \quad h_j(x) = 0, \quad j = 1, \dots, b \\ & \quad x \in \mathfrak{R}^l, \end{aligned} \tag{III.1}$$

Où x est appelé la variable d'optimisation, la fonction $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ est appelée la fonction objectif (ou fonction coût). Les équations $g_i(x) \leq 0$ sont appelées les contraintes inégalités et les équations $h_j(x) = 0$ sont appelées les contraintes égalités.

III.2.2. Les problèmes d'optimisation sans contraintes

Définition 1 :

On définit un problème d'optimisation sans contraintes par :

$$\begin{aligned} & \text{Minimiser } f(x) \\ & \quad x \in \mathfrak{R}^l \end{aligned} \tag{III.2}.$$

Définition 2 :

On dit que x^* est un minimum local si

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{V}(x^*)$$

On dit que x^* est un minimum global si

$$f(x^*) \leq f(x) \quad \forall x \in \mathfrak{R}^l$$

III.3 Concepts de base :**III.3.1 Gradient****Définition 3 :**

Soit f une fonction de \mathfrak{R}^l dans \mathfrak{R} admettant en un point x des dérivées partielles du premier ordre. On posera $x = (x_1, x_2, \dots, x_l)^T$

On note $\nabla f(x)$ et on appelle gradient de f au point x le vecteur-colonne :

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_l} \right)^T \quad (\text{III. 3})$$

Si $F(x) = (f_1(x), \dots, f_z(x))$ est un vecteur-ligne où (f_1, \dots, f_z) sont des fonctions réelles de l variables réelles dérivables au point x , alors $\nabla f(x)$ est la matrice dont la j^e colonne est $\nabla f_j(x)$.

III.3.2 Hessien**Définition 4 (la matrice hessienne)**

Si maintenant f admet des dérivées partielles d'ordre 2 en x , on pose

$$\nabla^2 f(x) = \nabla(\nabla f(x)^T),$$

$$\text{C'est-à-dire } \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_l} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_l} \\ \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_l \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial x_l^2} \end{bmatrix} \quad (\text{III.4})$$

$\nabla^2 f(x)$ s'appelle la matrice hessienne de f .

Si f est une fonction de classe C^2 (admet des dérivées partielles d'ordre 2 continues), la matrice hessienne de f est une matrice symétrique.

Théorème 1 : (condition nécessaire du premier ordre)

Soit $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ une fonction différentiable au point x^ si x^* est un minimum local alors*

$$\nabla f(x^*) = 0$$

Théorème 2 : (Condition nécessaire du second ordre)

Une condition nécessaire pour que x^ soit un minimum local ou global de f est :*

$$(a) \nabla f(x^*) = 0$$

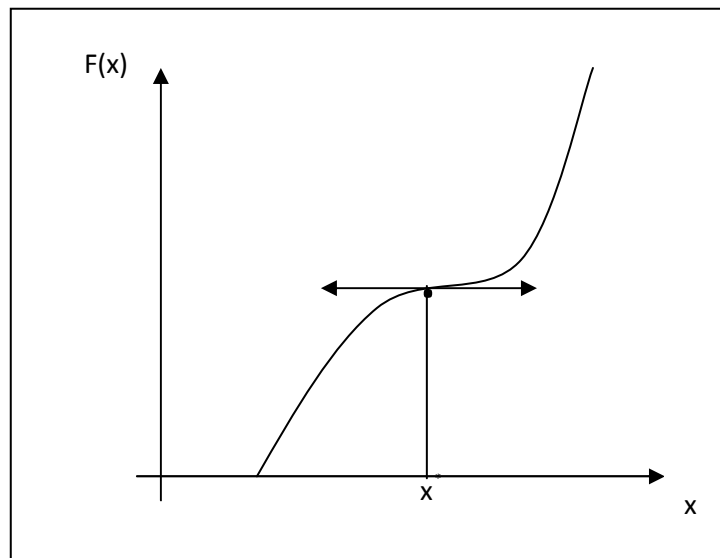
(b) la matrice hessienne $H(x^)$ est semi-définie positive.*

Remarque 1.

Ces conditions sont nécessaires non suffisantes.

Un point x^* qui vérifie la condition (a) c'est-à-dire $\nabla f(x^*) = 0$ est appelé un point stationnaire.

La figure(III.1) illustre le fait que la stationnarité n'est pas une condition suffisante d'optimalité locale.



Figure(III.1). x^* est un point stationnaire, mais ce n'est pas un minimum local.

Théorème 3 : (condition suffisante d'optimalité locale)

Soit $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ une fonction deux fois continûment différentiable, si x^* est un minimum local de f alors :

(a) $\nabla f(x^*) = 0$

(b) la matrice hessienne $H(x^*)$ est définie positive.

Les notions de maximum local et global sont définies de façon tout à fait similaire. Il suffit juste de ramener le problème de maximisation à un problème de minimisation, en écrivant

$$\max_x f(x) = -\min_x (-f(x))$$

Définition5 (direction de descente)

Supposons que $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ est différentiable au point x^* , s'il y a un vecteur p tel que

$$\nabla f(x)^T p < 0 \tag{III.5}$$

Et il existe un $\delta > 0$ tel que

$$f(x^* + \lambda p) < f(x^*) \tag{III.6}$$

pour tout $\lambda \in]0, \delta[$, alors p est une direction de descente.

III.3.3 La convexité

III.3.3.1 Ensembles convexes

Définition 6:

Un ensemble $S \in \mathfrak{R}^l$ est dit convexe si la ligne joignant chaque deux points dans l'ensemble S est toujours appartenue à cet ensemble. Autrement dit, un ensemble $S \subset \mathfrak{R}^l$ est convexe si $\forall x_1, x_2 \in S$ et $\lambda \in \mathfrak{R}$ on a $x = \lambda x_1 + (1 - \lambda)x_2 \in S$ pour tout $\lambda \in [0, 1]$

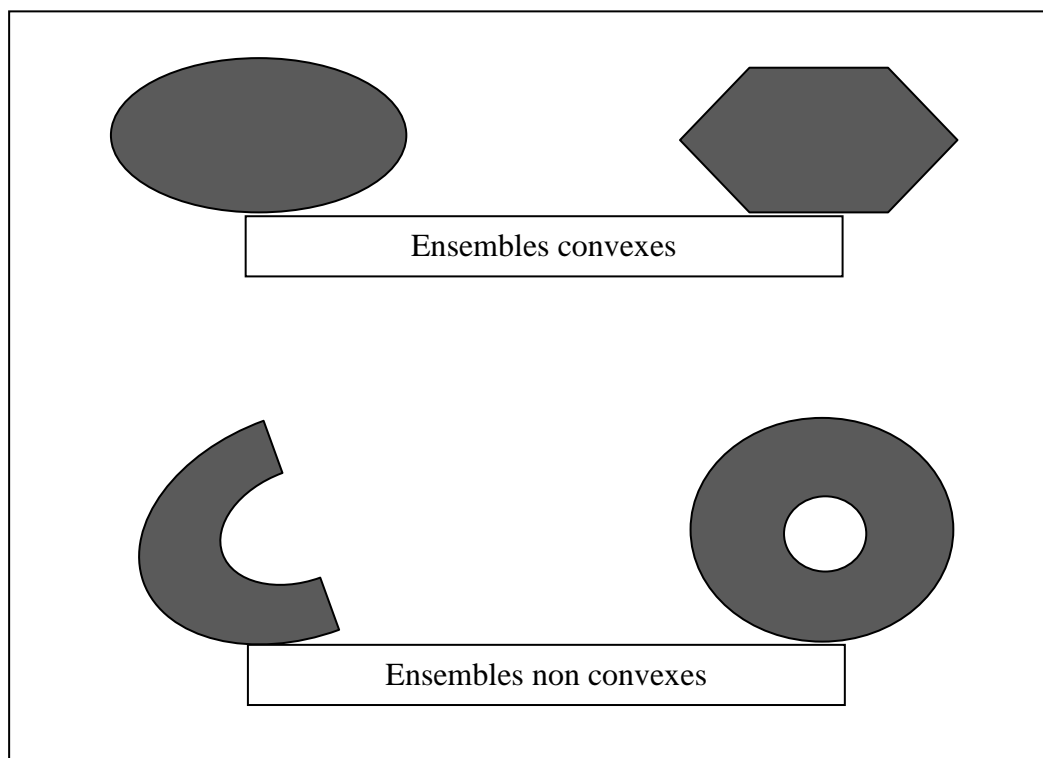


Figure III.2 : Ensembles convexes et non convexes

III.3.3.2 Fonctions convexes

Définition 7 :

Une fonction f définie sur un ensemble convexe S dans \mathfrak{R}^l est dite convexe si :

$$\forall x_1, x_2 \in S \text{ et } \lambda \in [0, 1] \\ f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \quad (\text{III.7})$$

f est dite strictement convexe sur S si l'inégalité est stricte pour $x_1 \neq x_2$, et $\lambda \in]0, 1[$.

III.3.3.2.1 Fonctions convexes différentiables

Définition 8 :

Si f est une fonction différentiable sur un ensemble convexe ouvert S dans \mathfrak{R}^l , alors f est convexe sur S si et seulement si pour chaque $x_1, x_2 \in S$ on a

$$f(x_1) - f(x_2) \geq \nabla f(x_1)^T (x_1 - x_2). \quad (\text{III.8})$$

III.3.3.2.2 Condition nécessaire et suffisante d'optimalité globale : [4]

Dans le cas d'une fonction convexe propre f définie sur \mathfrak{R}^l , une condition nécessaire et suffisante pour que x^* soit un minimum global de f est que 0 soit un sous gradient de f en x^* . Pour une fonction continûment différentiable, on obtient donc le théorème suivant :

Théorème 4 :

Soit $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ convexe continûment différentiable, une condition nécessaire et suffisante pour que x^* soit un optimum global de f sur \mathfrak{R}^l est que :

$$\nabla f(x^*) = 0$$

III.4 Méthode de Newton [4]

III.4.1 Introduction

La méthode de Newton est une méthode pour résoudre des problèmes d'optimisation sans contraintes de la forme

$$\begin{cases} \min_x f(x) \\ x \in \mathfrak{R}^l \end{cases} \quad (\text{III.9})$$

Où $f: \mathfrak{R}^l \rightarrow \mathfrak{R}$ est une fonction convexe et deux fois continûment différentiable.

En partant d'un point x^0 , l'algorithme de Newton va chercher à générer une suite d'itérés x^k défini par :

$$x^{k+1} = x^k + a^k p^k \quad (\text{III.10})$$

Où p^k est appelé le pas de Newton, et a^k est un scalaire positif appelé la longueur du pas.

III.4.2 Le pas de Newton

On définit le pas de Newton par :

$$p^k = -[\nabla_x^2 f(x^k)]^{-1} \nabla_x f(x^k) \quad (\text{III.11})$$

Puisque $\nabla_x^2 f(x^k)$ est défini positive, alors

$$\begin{aligned} [\nabla_x f(x^k)]^T p^k &= -[\nabla_x f(x^k)]^T [\nabla_x^2 f(x^k)]^{-1} \nabla_x f(x^k) < 0 \\ (\text{sauf si } \nabla_x f(x^k) &= 0 \end{aligned} \quad (\text{III.12})$$

Ceci implique que le pas de Newton est un pas de descente.

III.4.3 La longueur du pas

On choisit le pas a^k de telle sorte qu'on minimise la valeur de la fonction objectif f , c'est-à-dire :

$$f(x^k + a^k p^k) < f(x^k) \quad (\text{III.13})$$

Le choix de a^k se fait par la recherche linéaire, elle consiste à déterminer le pas a^k à effectuer le long d'un pas de descente p^k et les valeurs de ce pas peuvent être obtenues par les méthodes d'Armijo et Wolfe

III.4.3.1 Méthode d'Armijo :

La méthode d'Armijo consiste à éviter les pas trop grands, la condition de descente $f(x^k + a^k p^k) < f(x^k)$, n'est pas suffisante pour que la longueur du pas a^k soit considérée comme acceptable, de ce fait, on impose d'autres conditions : une première condition est dû à Armijo, cette condition stipule que a^k doit donner une diminution de la fonction objectif f , comme suit :

Soit c_1 une constante tel que $c_1 \in (0, 1)$

$$f(x^k + a^k p^k) \leq f(x^k) + c_1 a^k [\nabla_x f(x^k)]^T p^k \quad (\text{III.14})$$

Qu'on appelle l'inégalité d'Armijo.

La constante c_1 est choisie arbitrairement tel que $c_1 \in (0, 1)$ mais il est souvent préférable de choisir $c_1 < 0.5$

Comme cette condition n'est pas suffisante pour s'assurer que l'algorithme fait une progression raisonnable, On fait appel à une deuxième méthode qui est la méthode de Wolfe :

III.4.3.2 Méthode de Wolfe

Consiste à déterminer a^k vérifiant la condition (III.14) afin d'empêcher le pas a^k d'être trop petit, il lui impose de vérifier la condition supplémentaire

$$\nabla f(x^k + a^k p^k)^T p^k \geq c_2 \nabla f(x^k)^T p^k \quad (\text{III.15})$$

Où $0 < c_2 < c_1 < 1$.

III.4.3.3 Algorithme de recherche linéaire

Choisir $\bar{a} > 0$, $\eta \in (0, 1)$, $c \in (0, 1)$

Poser $a \leftarrow \bar{a}$

Répéter jusqu'à $f(x^k + ap^k) \leq f(x^k) + ca[\nabla_x f(x^k)]^T p^k$

$a \leftarrow \eta a$

Fin

$a_k \leftarrow a$

Remarque8

Dans la méthode de Newton, on choisit $\bar{a} = 1$.

III.4.4 Algorithme de Newton

Choisir $x^0 \in \mathfrak{R}^l$, une tolérance $\varepsilon > 0$

$k \leftarrow 0$

Tant que $\|\nabla_x f(x^k)\| > \varepsilon$ faire

1. Calculer le pas de Newton

$$p^k = -[\nabla_x^2 f(x^k)]^{-1} \nabla_x f(x^k)$$

2. Choisir a^k par la recherche linéaire

3. Mise à jour

$$x^{k+1} = x^k + a^k p^k$$

$k \leftarrow k + 1$

Fin.

Remarque 9

Une propriété importante de la méthode est qu'elle converge en une seule itération lorsqu'elle est appliquée à une fonction strictement convexe.

Exemple d'application

Considérons le problème d'optimisation sans contraintes suivant :

$$\text{Minimiser } f(x_1, x_2) = x_1^2 + x_2^2 - x_1 x_2$$

$$x \in \mathfrak{R}^2$$

$$\nabla_x f(x^k) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 \\ 2x_2 - x_1 \end{bmatrix}$$

$$\nabla_x^2 f(x^k) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Puisque les mineurs principaux d'ordre 1 et d'ordre 2 $\begin{cases} |2| \\ \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} \end{cases}$ sont positifs alors $f(x)$ admet un minimum.

Itération 0

$$x^0 = (1,1)$$

$$\nabla_x f(x^0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Calcul du pas de Newton p^0

$$p^0 = - \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Calcul de la longueur du pas a^0

On pose $\eta = \frac{1}{2}$, $c = 0.1$, $\bar{a} = 1$

$$f(x^0 + ap^0) = f[(1,1) + (-1, -1)] = f(0,0) = 0$$

$$f(x^0) = f(1,1) = 1$$

$$\nabla_x f(x^0)^T p^0 = [1,1] \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -2$$

$$f(x^0) + ac \nabla_x f(x^0)^T p^0 = 1 + 0.1(-2) = 0.8$$

$0.8 > 0$ alors la condition est remplie

Donc $a = 1$ est la longueur du pas.

La mise à jour

$$x^1 = x^0 + p^0 = (1,1) + (-1, -1) = (0,0)$$

Test d'optimalité

$$\nabla_x f(x^1) = (0,0) \text{ donc } x^1 \text{ est optimal.}$$

III.5 Conclusion

Dans ce chapitre, nous avons présenté quelques notions de l'optimisation des fonctions mathématiques. Nous avons aussi présenté la méthode de Newton permettant de résoudre itérativement les problèmes d'optimisations. Cette méthode sera utilisé dans le chapitre suivant pour résoudre le problème de commande optimale une fois il est transformé en un problème d'optimisation.

IV.1 Introduction

Dans ce chapitre, on présente une approche de calcul de la commande optimale pour système dynamique. L'approche consiste à définir la loi de commande comme une somme de fonctions temporelles connues pondérées par des coefficients. Ces derniers sont déterminés de manière à optimiser le critère de performance. Cette forme de la commande permet de ramener le problème de la commande optimale à un problème d'optimisation en utilisant la méthode de décomposition d'Adomian.

IV.2 Principe de l'approche Indirecte[5]

Dans cette approche indirecte, on considère que la loi de commande est de la forme :

$$u(t) = \sum_{j=0}^M k_j \theta_j(t) \quad (\text{IV. 1})$$

Où les $\theta_j(t)$ sont des fonctions temporelles pondérées par des paramètres de réglage k_j

Le problème s'agit de trouver

$$\left\{ \begin{array}{l} \min_{u(t)} J(u(t)) = \int_0^{t_f} G(x(t), u(t), t) dt \\ \text{S. à} \\ \dot{x}(t) = f(x(t), u(t), t) \\ x(0) = x_0 \end{array} \right. \quad (\text{IV. 2})$$

Où J est le critère de performance à minimiser et $x(t)$ est la variable d'état.

IV.2.1 Procédure

On prend $\theta_j(t) = t^j, j = 0, \dots, M$

$$\text{D'où } \begin{cases} \theta_0(t) = 1 \\ \theta_1(t) = t \\ \theta_2(t) = t^2 \\ \vdots \\ \theta_N(t) = t^M \end{cases}$$

Et à partir de l'équation (IV.1), on aura l'expression de la commande comme suit :

$$u(t) = k_0 + k_1 t + k_2 t^2 + \dots + k_M t^M \quad (\text{IV.3})$$

On remplace l'expression de la commande $u(t)$ dans l'équation d'état $\dot{x}(t)$, on aura :

$$\dot{x}(t) = f(x(t), k_0 + k_1 t + k_2 t^2 + \dots + k_M t^M) \quad (\text{IV.4})$$

Pour résoudre cette équation, on utilise la méthode de décomposition d'Adomian, présentée dans le chapitre 2, et qui donne la solution $x(t)$ sous forme

$$x(t) = F(k_0, k_1, k_2, \dots, k_M, t) \quad (\text{IV.5})$$

Une fois la solution $x(t)$ est obtenue, on la remplace dans l'équation du critère de performance « J » à minimiser.

D'une part, on a :

$$J = \int_0^{t_f} G(x(t), u(t), t) dt \quad (\text{IV.6})$$

Et d'autre part :

$$\begin{cases} x(t) = F(k_0, k_1, k_2, \dots, k_M, t) \\ u(t) = k_0 + k_1 t + k_2 t^2 + \dots + k_M t^M \end{cases} \quad (\text{IV.7})$$

En remplaçant les équations (IV.7) dans (IV.6), on obtient le problème d'optimisation sans contraintes suivant :

$$\begin{aligned} \min_k J &= \int_0^{t_f} G(k_0, k_1, k_2, \dots, k_M, t) dt, \\ &= H(k_0, k_1, k_2, \dots, k_M) \end{aligned}$$

$$k \in \mathfrak{R}^{M+1}$$

Et le problème revient à minimiser le critère de performance « J » par rapport aux paramètres « k_j »

$$\min_{k_j} J = H(k_0, k_1, k_2, \dots, k_M) \quad (\text{IV.8})$$

Expression sur laquelle nous allons appliquer la méthode de Newton pour calculer les paramètres « $k_0, k_1, k_2, \dots, k_M$ » afin de trouver l'expression de la commande optimale $u^*(t)$.

IV.3 Application :

Soit le système décrit comme suit :

$$\left\{ \begin{array}{l} \min J = \int_0^{t_f} x^2(t) + u^2(t) \\ \text{S. à} \\ \dot{x} = -x + u \\ x(0) = 1 \end{array} \right. \quad (\text{IV.9})$$

Sachant que :

$$u = \sum_{j=0}^M k_j \theta_j(t)$$

$$\text{Et } \theta_j(t) = t^j$$

En limitant l'ordre à « M= 2 », on aura :

$$u(t) = k_0 + k_1 t + k_2 t^2 \quad (\text{IV.10})$$

D'autre part

$$\dot{x} = -x + u \quad (\text{IV.11})$$

En remplace (IV.10) dans (IV.11), on aura

$$\dot{x} + x = k_0 + k_1 t + k_2 t^2 \quad (\text{IV.12})$$

D'après la méthode de décomposition d'Adomian, l'équation (IV.12) peut s'écrire comme suit :

$$Lx(t) + x(t) = k_0 + k_1 t + k_2 t^2 \quad (\text{IV.13})$$

Tel que :

$$L = \frac{d}{dt}$$

Et son inverse :

$$L^{-1} = \int_0^t (.) dt$$

On introduit l'opérateur L^{-1} à l'équation (IV.13) , on aura :

$$L^{-1}Lx(t) + L^{-1}x(t) = L^{-1}(k_0 + k_1t + k_2t^2)$$

1. On calcul $L^{-1}Lx(t)$:

$$L^{-1}Lx(t) = \int_0^t Lx(t)dt$$

$$L^{-1}Lx(t) = \int_0^t \dot{x}(t)dt$$

$$L^{-1}Lx(t) = x(t) - x(0)$$

$$L^{-1}Lx(t) = x(t) - 1$$

2. Le calcul de $L^{-1}(k_0 + k_1t + k_2t^2)$:

$$L^{-1}(k_0 + k_1t + k_2t^2) = \int_0^t (k_0 + k_1t + k_2t^2)dt$$

$$L^{-1}(k_0 + k_1t + k_2t^2) = k_0t + \frac{1}{2}k_1t^2 + \frac{1}{3}k_2t^3$$

D'où

$$x(t) = -L^{-1}x(t) + 1 + k_0t + \frac{1}{2}k_1t^2 + \frac{1}{3}k_2t^3$$

Avec

$$x(t) = \sum_{n=0}^{\infty} x_n(t)$$

On aura :

$$\sum_{n=0}^{\infty} x_n(t) = -L^{-1}\left(\sum_{n=0}^{\infty} x_n(t)\right) + 1 + k_0t + \frac{1}{2}k_1t^2 + \frac{1}{3}k_2t^3$$

Et le schéma itératif est donné comme suit :

$$\begin{cases} x_0(t) = 1 + k_0 t + \frac{1}{2} k_1 t^2 + \frac{1}{3} k_2 t^3 \\ x_{k+1}(t) = -L^{-1} x_k(t) \end{cases}$$

Pour $k = 0$:

$$x_1(t) = -L^{-1} x_0(t)$$

$$x_1(t) = - \int_0^t (1 + k_0 t + \frac{1}{2} k_1 t^2 + \frac{1}{3} k_2 t^3) dt$$

$$x_1(t) = - \left[t + \frac{1}{2} k_0 t^2 + \frac{1}{3} k_1 t^3 + \frac{1}{12} k_2 t^4 \right]_0^t$$

$$x_1(t) = -t - \frac{1}{2} k_0 t^2 - \frac{1}{3} k_1 t^3 - \frac{1}{12} k_2 t^4$$

Pour $(k = 1)$:

$$x_2(t) = -L^{-1} x_1(t)$$

$$x_2(t) = - \int_0^t \left(-t - \frac{1}{2} k_0 t^2 - \frac{1}{3} k_1 t^3 - \frac{1}{12} k_2 t^4 \right) dt$$

$$x_2(t) = \frac{1}{2} t^2 + \frac{1}{6} k_0 t^3 + \frac{1}{24} k_1 t^4 + \frac{1}{60} k_2 t^5$$

Pour $k = 2$:

$$x_3(t) = -L^{-1} x_2(t)$$

$$x_3(t) = - \int_0^t \left(\frac{1}{2} t^2 + \frac{1}{6} k_0 t^3 + \frac{1}{24} k_1 t^4 + \frac{1}{60} k_2 t^5 \right) dt$$

$$x_3(t) = -\frac{1}{6} t^3 - \frac{1}{24} k_0 t^4 - \frac{1}{120} k_1 t^5 - \frac{1}{360} k_2 t^6$$

⋮

D'après la formule $\phi_n(t) = \sum_{n=0}^{+\infty} u_n(t)$, on trouve

$$\begin{aligned} \phi_n(t) = & \left[-t - \frac{1}{2} k_0 t^2 - \frac{1}{3} k_1 t^3 - \frac{1}{12} k_2 t^4 \right] + \left[\frac{1}{2} t^2 + \frac{1}{6} k_0 t^3 + \frac{1}{24} k_1 t^4 + \frac{1}{60} k_2 t^5 \right] \\ & + \left[-\frac{1}{6} t^3 - \frac{1}{24} k_0 t^4 - \frac{1}{120} k_1 t^5 - \frac{1}{360} k_2 t^6 \right] + \dots \end{aligned}$$

$$\begin{aligned}\phi_n(t) = & \left(1 - t + \frac{1}{2}t^2 - \frac{1}{3!}t^3 + \dots\right) + k_0 \left(t - \frac{1}{2}t^2 + \frac{1}{3!}t^3 - \frac{1}{4!}t^4 + \dots\right) \\ & + k_1 \left(\frac{1}{2}t^2 - \frac{1}{3!}t^3 + \frac{1}{4!}t^4 - \frac{1}{5!}t^5 + \dots\right) + k_2 \left(\frac{1}{3}t^3 - \frac{1}{12}t^4 + \frac{1}{60}t^5 - \frac{1}{360}t^6 + \dots\right)\end{aligned}$$

En utilisant les développements aux limites:

$$e^{-t} = 1 - t + \frac{t^2}{2!} - \frac{t^3}{3!} + \frac{t^4}{4!} - \dots$$

$$sh(t) = \sum_{n=0}^{\infty} \frac{t^{2n+1}}{(2n+1)!} = t + \frac{t^3}{3!} + \frac{t^5}{5!} + \dots$$

Et

$$ch(t) = \sum_{n=0}^{\infty} \frac{t^{2n}}{(2n)!} = 1 + \frac{t^2}{2!} + \frac{t^4}{4!} + \dots$$

On aura

$$x(t) = e^{-t} + k_0(-e^{-t} + 1) + k_1(e^{-t} - 1 + t) + 2k_2 \left(-e^{-t} + 1 - t + \frac{1}{2}t^2\right)$$

Une fois la solution $x(t)$ est obtenue, on remplace son expression ainsi que l'expression de la commande $u(t)$ dans l'équation du critère de performance donnée par la formule (IV.9). Nous appliquons la méthode de Newton pour identifier les paramètres de vecteur de commande « k_0, k_1, k_2 ».

Le problème d'optimisation final obtenu est résolu par la Méthode de Newton, en considérant une solution de départ $k^0 = [0, 0, 0]$, après une seule itération, la méthode converge et donne les résultats suivant :

$$k_0 = -0.4089$$

$$k_1 = 0.7827$$

$$k_2 = -0.3219$$

D'où la commande optimale est

$$u^*(t) = -0.4089 + 0.7827t + -0.3219t^2$$

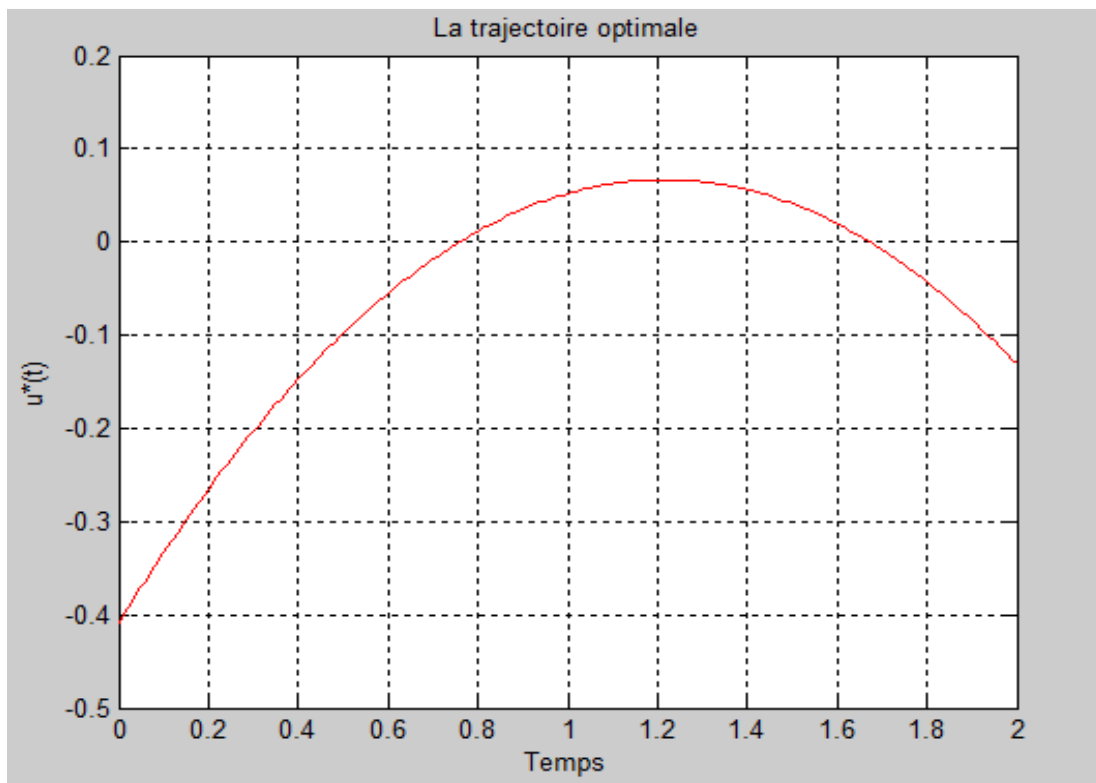
La variable d'état est :

$$x(t) = 2.8354e^{-t} - 0.3219t^2 + 1.4265t - 1.8354$$

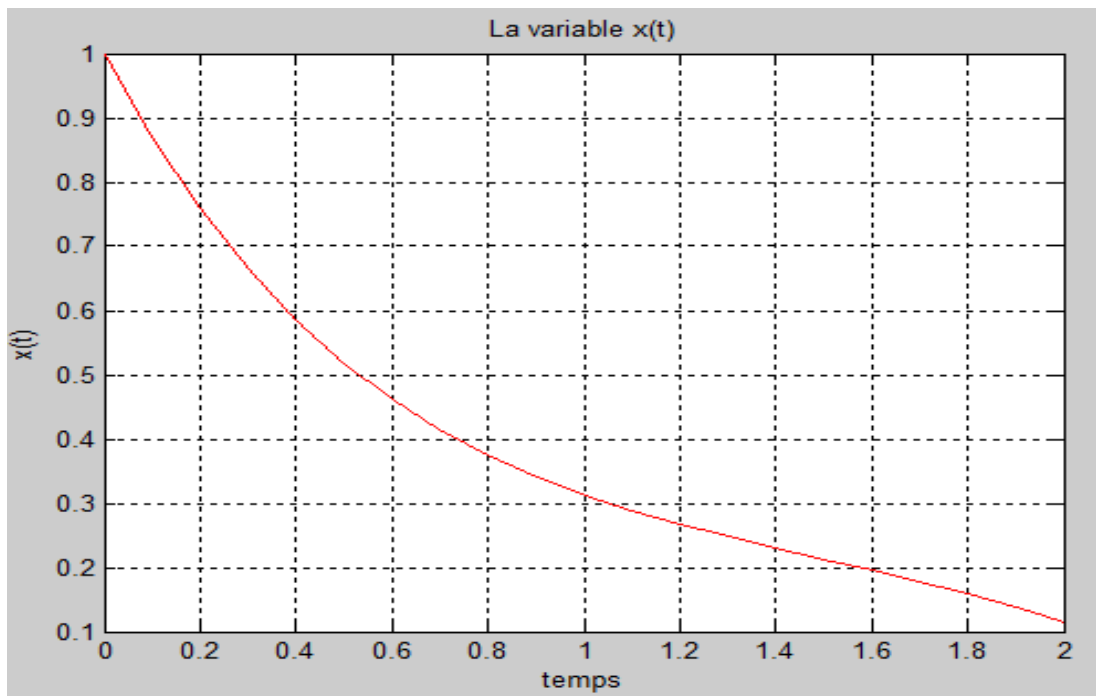
Et la valeur du critère de performance est

$$J = 0.0425$$

Les trajectoires de la commande optimale ainsi de la variable d'état du système sont représentées respectivement par les figures (IV.1) et (IV.2)



Figure(IV.1) : La trajectoire de la commande optimale



Figure(IV.2) : La trajectoire de la variable d'état

IV.4 Conclusion

Dans ce chapitre, on a présenté une démarche pour transformer le problème de commande optimale en un problème d'optimisation en utilisant la méthode de décomposition d'Adomian. Le principe consiste à exprimer la variable de commande comme une somme de fonctions temporelles pondérées par des paramètres de réglage. Ces derniers sont déterminés par optimisation.

Le travail présenté dans ce mémoire concerne la résolution de problèmes de commande optimale en utilisant l'approche indirecte. Elle consiste à transformer le problème de commande optimale en un problème d'optimisation. Pour ce faire une méthode de résolution des équations différentielles appelé méthode de décomposition d'Adomian.

Ainsi, dans un premier temps, nous avons introduit les notions et concepts de base pour la formulation d'un problème de commande optimale. Par la suite, on a exposé et illustré par des exemples la méthode d'Adomian utilisée pour résoudre les équations différentielles. Des notions d'optimisation numérique de fonctions en utilisant la méthode de Newton ont été présentées suivi de la démarche à suivre pour résoudre un problème de commande optimale par la méthode de décomposition d'Adomian combinée avec la méthode de Newton.

La méthode de décomposition d'Adomian permet de ramener un problème de commande optimale à un problème d'optimisation qui sera ensuite résolu par la méthode de Newton. L'idée consiste à définir la loi de commande par une série de fonctions pondérées par des paramètres de réglage. Ces fonctions sont en fonction du temps et connues, le problème revient à chercher les paramètres de pondération permettant de minimiser le critère de performance. En utilisant la méthode d'Adomian et en faisant certaines manipulations mathématiques, on obtient un problème d'optimisation dont les variables d'optimisation sont les pondérations. L'approche présentée a été illustrée par exemple d'application.

L'approche proposée peut être étendue pour prendre en compte des contraintes du type inégalités sur les variables d'état et/ou de commandes.

Fonctions	Développements limités
e^x	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$
$\sin x$	$\frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$
$\cos x$	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$
shx	$\frac{x}{1!} + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots$
chx	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots$
$\frac{1}{1-x}$	$1 + x + x^2 + x^3 + x^4 + \dots$
$\frac{1}{1+x}$	$1 - x + x^2 - x^3 + x^4 - \dots$

Indices

$x(t)$: La variable d'état

$x(t_0)$: État initial

$x(t_f)$: État final

t_0 : Instant initial

t_f : Instant final

T : Horizon

l : Nombre de variables d'état

p : Nombre de variables de commande

r : Nombre de sorties

$u(t)$: La commande

$u^*(t)$: La commande optimale

J : Critère de performance

Ψ : Fonction scalaire

G : Fonctionnelle

L, R : Des opérateurs linéaires

N : Opérateurs non linéaire

A_n : Les polynômes d'Adomian

$\nabla f(x)$: Le Gradient

$\nabla^2 f(x)$: Le Hessien

p^k : Le pas de Newton

a^k : La longueur du pas de Newton

c_1 : La constante d'Armijo

c_2 : La constante de Wolfe

$\theta_j(t)$: Fonctions temporelles

k_j : Paramètres de pondération

BIBLIOGRAPHIE

- [1] : **D.E . Kirk**, Optimal control theory. An Introduction. Prentice-Hall, 1970.
- [2] **B. O. KONFE**, Nouvelles méthodes mathématiques, Alienor et Adomian, pour la biomédecine, Thèse de Doctorat, Université de Ouagadougou, Boukina Faso, 2005.
- [3] **M. Bregounioux**, Optimisation sans contraintes, Département de mathématique, Orléans, 2003-2004
- [4] **A. Rondepierre et P. Weiss**, Méthodes standards en optimisation non linéaires déterministes, Toulouse, 2012-2013.
- [5] **J.P. Couriou**, Commande des Procédés, 2^{ième} édition, Lavoisier, 2003.
- [6]: **A. E. Bryson**, Dynamic Optimization. Addison Wesley, 1999.
- [7]: **D. S. Naidu**, Optimal control systems; CRC Press, Boca Raton, Florida, 2003.
- [8]: **G. Adomian**. Solving frontier problems of physics: The decomposition method. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [9]: **D. P. BertsiKas**, Nonlinear Programming, Second edition, Athena Scientific, 1999.
- [10] **S. Boyd, L. Vandenberghe**, Convex optimization, Cambridge University Press, 2004.
- [11]: **M. S. Bazaraa**, Nonlinear programming, Theory and algorithms, Third edition Wiley, 2006.

Résumé

Le travail présenté dans ce mémoire concerne la résolution de problèmes de commande optimale en utilisant l'approche indirecte. Elle consiste à transformer le problème de commande optimale en un problème d'optimisation. Pour ce faire une méthode de résolution des équations différentielles appelé méthode de décomposition d'Adomian.

La méthode de décomposition d'Adomian permet de ramener un problème de commande optimale à un problème d'optimisation qui sera ensuite résolu par la méthode de Newton. L'idée consiste à définir la loi de commande par une série de fonctions pondérées par des paramètres de réglage. Ces fonctions sont en fonction du temps et connues, le problème revient à chercher les paramètres de pondération permettant de minimiser le critère de performance.