

10/91

20

وزارة الجامعات
Ministère aux Universités

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

SUJET

COMMANDE ADAPTATIVE A MODELE
DE REFERENCE DES ROBOTS
MANIPULATEURS

Proposé par :

B.BOUZOUIA

Etudié par :

LAHMER
Toufik

Dirigé par :

B.BOUZOUIA

PROMOTION
Juin 1991

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وزارة الجامعات
Ministère aux Universités

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

SUJET

COMMANDE ADAPTATIVE A MODELE
DE REFERENCE DES ROBOTS
MANIPULATEURS

Proposé par :

B. BOUZOUIA

Etudié par :

LAHMER

Toufik

Dirigé par :

B. BOUZOUIA

PROMOTION

Juin 1991

A mes parents,

A mes frères,

A ma soeur,

A ma famille,

A mes amis,

A mon pays.

REMERCIEMENTS

Merci à celui qui m' a appris ce que j'ignorais.

Merci à :

- Monsieur O.Stihi responsable de la scolarité
d'Automatique, qui a tout fait pour une bonne marche des
travaux.

- Monsieur A.Bensenouci chef de département du Génie
Electrique,

- Messieurs les membres du jury, pour avoir accepté de
l'être,

- tous mes enseignants qui ont participé à mon éducation.

Je tiens à remercier plus particulièrement Monsieur
B.Bouzouia, Docteur en Robotique de l'université Paul-Sabatier
de Toulouse, pour sa proposition de ce sujet, pour l'encadrement
de mes travaux et pour ses conseils utiles et ininterrompus.

Je remercie aussi Hammoud, Kamel, Karim, Laredj, Mohammed, Nacer
et la section d'Electrotechnique pour leur réussite dans la
création d'une ambiance de travail. Ainsi que tous ceux qui m'ont
aidé matériellement ou moralement.

SOMMAIRE



INTRODUCTION GENERALE	1
CHAPITRE I : MODELISATION DES ROBOTS MANIPULATEURS	
I-1- Quelques définitions	4
I-2- Modèle dynamique	5
I-3- Représentation d'état	10
CHAPITRE II : COMMANDE ADAPTATIVE A MODELE DE REFERENCE DES ROBOTS MANIPULATEURS	
II-1- Introduction	14
II-2- MRAS basée sur l'optimisation locale des paramètres	16
II-3- MRAS basée sur la stabilisation à l'aide de la théorie de Lyapunov	19
II-4- MRAS basée sur la stabilisation à l'aide de la théorie de Popov	23
CHAPITRE III : SIMULATION	
III-1- Introduction	29
III-2- Méthodes d'intégration	31
III-3- Cas du robot double pendule	33
III-4- Cas du robot TH8 à trois liaisons	35
III-5- Choix du modèle de référence	37
III-6- Génération de la trajectoire	38
III-7- Schéma global de simulation	41
CHAPITRE IV : PRESENTATION DES RESULTATS	
IV-1- Suivi de trajectoire la méthode de Lyapunov ..	48
IV-2- Suivi de trajectoire la méthode de Popov.....	49
CONCLUSION GENERALE	53
BIBLIOGRAPHIE	55
ANNEXES.....	57

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Introduction

INTRODUCTION GENERALE

La commande des robots manipulateurs a connu plusieurs évolutions depuis son apparition. Au début elle était classique, elle reposait sur l'automatique linéaire i.e. chaque liaison est représentée par un modèle linéaire et asservie indépendamment des autres liaisons. Cette méthode a certes remporté à l'industrie un succès considérable grâce à sa facilité d'implantation. Mais la nécessité d'effectuer des déplacements rapides et précis, a imposé la prise en considération de la dynamique du robot manipulateur. Dans cette méthode appelée commande dynamique, le robot manipulateur est représenté par des équations différentielles du second ordre non-linéaires. Cette méthode peut être fondée sur l'un des trois principes suivants :

- _ le calcul de l'effort généralisé,
- _ le découplage non-linéaire,
- _ et la linéarisation du modèle dynamique autour du mouvement de consigne.

Cependant, lors de l'utilisation du modèle identifiée, il s'est avéré que le temps de calcul était important. Pour cela la commande des robots manipulateurs a été dirigée vers la commande adaptative. Le problème rencontré par les chercheurs dans ce domaine est celui de la non stabilité du système commandé. D'ailleurs c'est à cause du fameux dilemme de l'automatique (stabilité-précision). Ce type de commande connaît deux grandes classes :

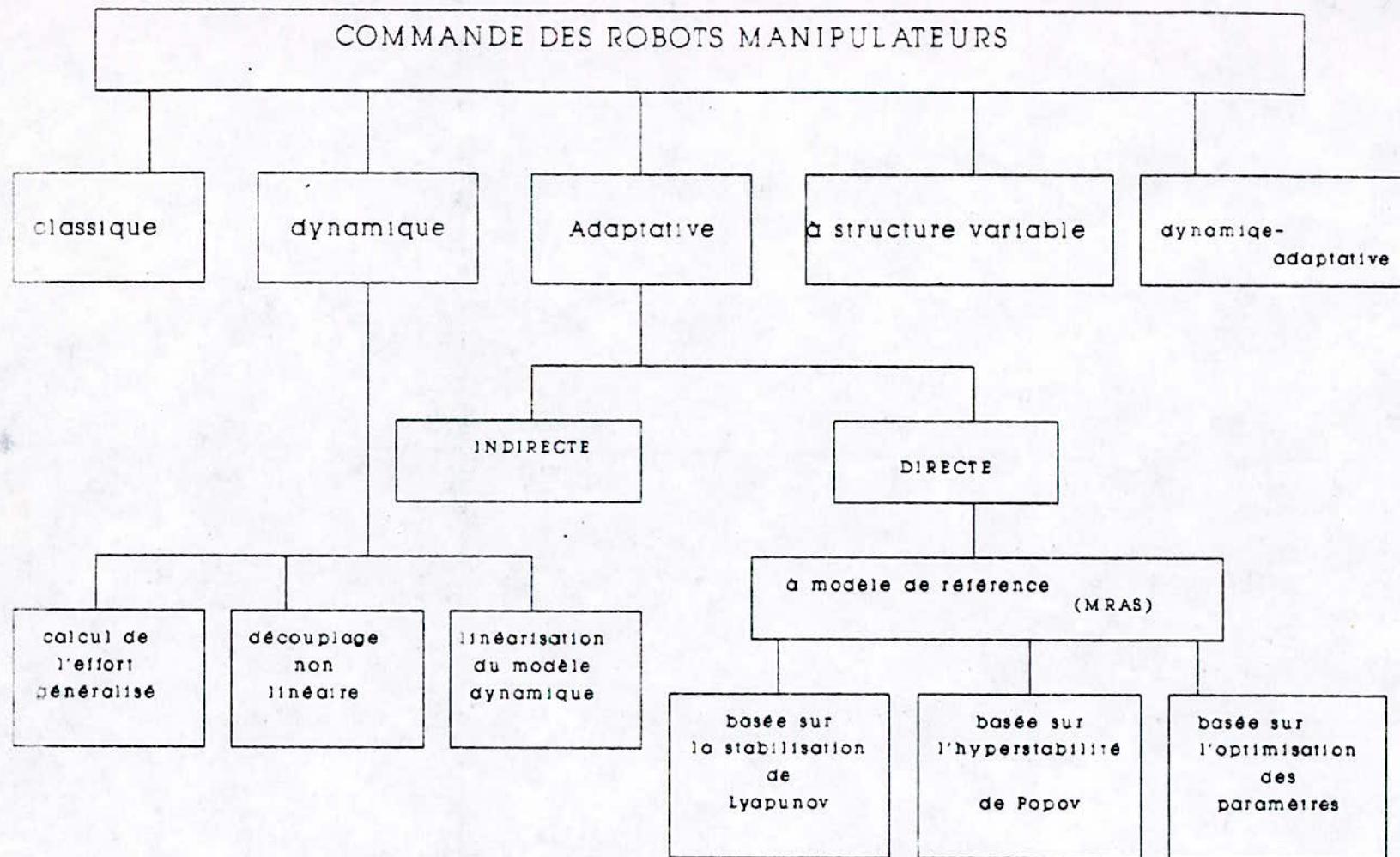
- commande adaptative indirecte, qui repose sur l'identification du système à chaque instant, ensuite remplacer les paramètres réels du système par leurs estimés, dans le calcul de la commande.

- commande adaptative directe, qui consiste à identifier le système de commande. La forme la plus répandue de ce type de commande est celle à modèle de référence, qu'on va étudier dans ce mémoire.

La combinaison des deux types précédents a donnée naissance à un autre type de commande qualifié d'adaptatif-dynamique.

Par ailleurs, il existe un autre type de commande dit à structure variable, qui consiste à décomposer le système en sous-systèmes continus entre eux; auxquels est associée une logique de commutation adéquate.

Notre travail consiste à implanter un logiciel de simulation de commande adaptative à modèle de référence (MRAS:model reference adaptive system) appliquée aux robots manipulateurs, et ceci en utilisant quelques algorithmes d'adaptation.



DIFFERENTS TYPES DE COMMANDE

DES ROBOTS MANIPULATEURS

Chapitre I

CHAPITRE I

MODELISATION DES ROBOTS MANIPULATEURS

I-1 - Quelques définitions:

.Espace généralisé:

C'est l'espace dans lequel est définie la configuration du robot manipulateur (R.M.). Si le robot est à structure de chaîne cinématique simple (i.e. constituée de corps liés les uns aux autres par des liaisons rotoïdes ou prismatiques), sa configuration est définie par la matrice colonne q constituée des n coordonnées généralisées qui spécifient les valeurs des mouvements élémentaires q_i aux différentes liaisons:

$$q = [q^1, \dots, q^n]^T$$

on définit aussi la matrice des vitesses généralisées:

$$\dot{q} = [\dot{q}^1, \dots, \dot{q}^n]^T$$

et la matrice des accélérations généralisées:

$$\ddot{q} = [\ddot{q}^1, \dots, \ddot{q}^n]^T$$

A cet espace est associée la commande généralisée.

.Espace opérationnel:

C'est l'espace dans lequel on définit la situation (position et orientation) de l'organe terminal du R.M. On définit alors:

.la matrice des coordonnées opérationnelles:

$$x = [x_1, \dots, x_m]^T$$

.la matrice des vitesses opérationnelles:

$$\dot{x} = [\dot{x}_1, \dots, \dot{x}_m]^T$$

.la matrice des accélérations opérationnelles:

$$\ddot{x} = [\ddot{x}_1, \dots, \ddot{x}_m]^T$$

$$\ddot{\mathbf{x}} = [\ddot{x}_1, \dots, \ddot{x}_m]^t$$

avec $m \geq 6$.

A cet espace est associée la commande opérationnelle.

I-2 - Modèle dynamique:

Le R.M. étant supposé à structure de chaîne cinématique simple, sa configuration dans l'espace généralisé est définie par la matrice colonne q des coordonnées généralisées.

Le modèle dynamique permet d'obtenir les couples (pour les liaisons rotoïdes) et les forces (pour les liaisons prismatiques) aux liaisons en fonction des coordonnées, vitesses et accélérations généralisées engendrées [1] :

$$\Gamma = f(q, \dot{q}, \ddot{q})$$

où : $\Gamma = [\Gamma^1, \dots, \Gamma^n]^t$ est le vecteur des efforts généralisés aux différentes liaisons.

L'expression de l'effort généralisé Γ^i pour la $i^{\text{ème}}$ liaison est:

$$\Gamma^i = \sum_{j=1}^n a^{ij}(q) \ddot{q}^j + \sum_{j,k=1}^n b^{i,jk}(q) \dot{q}^j \dot{q}^k - G^i(q) + F^{ii} \dot{q}^i + H^{ii} \text{sgn}(\dot{q}^i) \quad (I.1)$$

Γ^i s'exerçant sur le $i^{\text{ème}}$ corps, on a:

$$- \sum_{j=1}^n a^{ij}(q) \ddot{q}^j$$

représente l'effort d'inertie d'accélération,

$$- \sum_{j,k=1}^n b^{i,jk}(q) \dot{q}^j \dot{q}^k$$

représente les efforts d'inertie centrifuge et de coriolis,

$$G^i(q)$$

représente l'effort de pesanteur,

$$-F^{ii}(\dot{q}^i)$$

représente l'effort de frottement visqueux,

$$-H^{ii} \operatorname{sgn}(\dot{q}^i)$$

représente l'effort de frottement sec;

où: $b^{i,jk}$ sont les symboles de christoffel de la matrice d'énergie cinétique donnés par:

$$b^{i,jk} = \frac{1}{2} \left[\frac{\partial a^{ij}}{\partial q^k} + \frac{\partial a^{ik}}{\partial q^j} - \frac{\partial a^{jk}}{\partial q^i} \right]$$

Dans le cas multivariable, le modèle dynamique s'écrit sous la forme matricielle suivante:

$$\Gamma = A(q)\ddot{q} + B(q,\dot{q}) - g(q) + F\dot{q} + H \operatorname{sgn}(\dot{q}) \quad (\text{II.2})$$

avec:

$A(q) = A$ est la matrice (des a^{ij}) d'énergie cinétique, toujours régulière.

$$B(q,\dot{q}) = B = [b^1, \dots, b^n]^t$$

où $b^i = \dot{q}^t B^i \cdot \dot{q}$ est une forme quadratique des vitesses généralisées et où B^i a pour éléments d'indices j et k : $b^{i,jk}$.

Cette matrice $B(q,\dot{q})$ peut se mettre sous l'une des deux formes suivantes:

1. $B(q, \dot{q}) = C(q) \cdot h(\dot{q})$ avec $h(\dot{q}) = [\dot{q}^1 \dot{q}^1 \dots \dot{q}^1 \dot{q}^n \dot{q}^2 \dot{q}^2 \dots \dot{q}^n \dot{q}^n]^T$
 et les éléments de $C(q)$ sont les doubles des symboles de christofell de $A(q)$.

2. $B(q, \dot{q}) = B'(q, \dot{q}) \cdot \dot{q}$ où $B'(q, \dot{q})$ est une matrice $(n \times n)$ à expression non unique.

$G(q) = G'(q) \cdot q$ est la matrice des G^i .

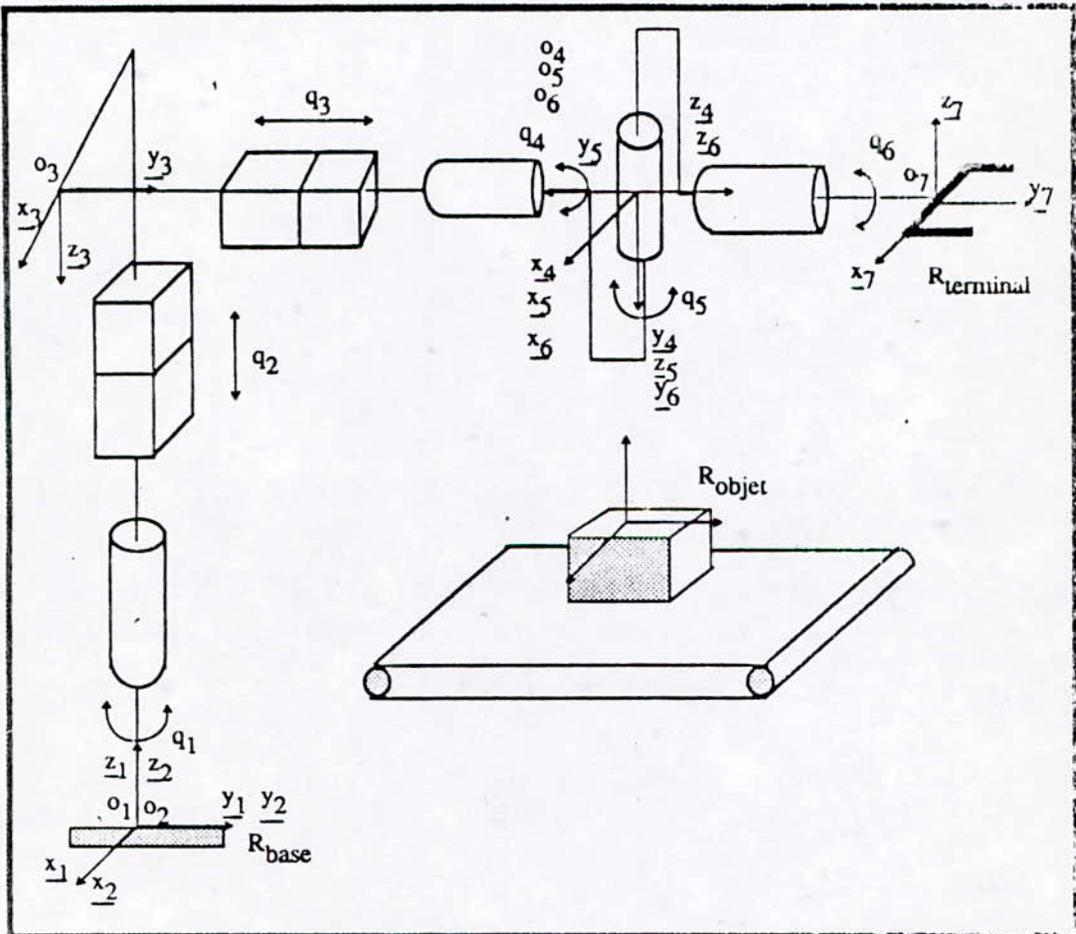


fig.I.1. Chaîne cinématique simple du robot T.H.8.

Dont les caractéristiques sont:

i	1	2	3	4	5	6
q^i_{min}	-135°	0.81 m	0.632 m	$-178,5^\circ$	-105°	$-180^\circ + \epsilon$
q^i_{max}	$+135^\circ$	1.31 m	1.432 m	$+178,5^\circ$	$+105^\circ$	$+180^\circ - \epsilon$

D'une manière générale, l'équation dynamique du R.M. est combinée à l'équation dynamique régissant le fonctionnement de l'ensemble Amplificateur-Actionneur-Réducteur. Et si l'actionneur est commandé en courant, le schéma est le suivant:

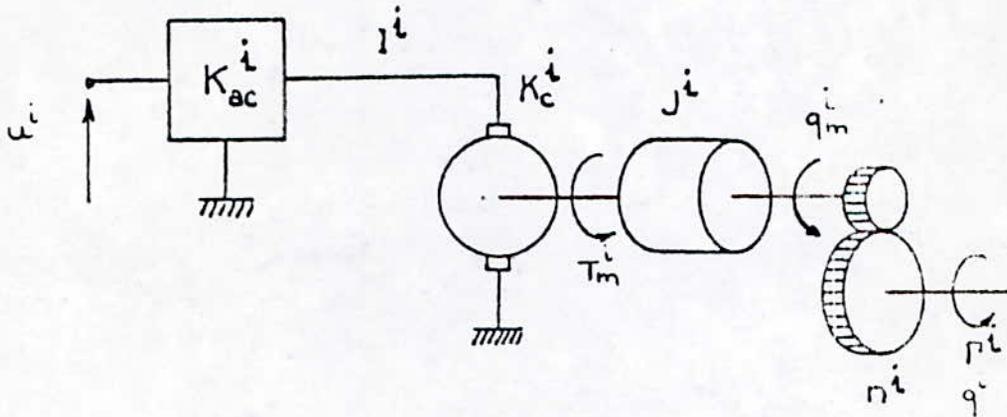


fig.II.1. Amplificateur-Actionneur-Réducteur.

où: u^i est la tension de commande,
 K_{ac}^i est le gain de l'amplificateur tension-courant,
 $I^i = K_{ac}^i \cdot u^i$ est le courant moteur,
 K_c^i est la constante du couple moteur,
 $T_m^i = K_c^i$ est le couple moteur,
 J^i est l'inertie propre du moteur et du reducteur,
"vue" par le moteur,

q_m^i est la position angulaire de l'axe moteur,

n^i est le rapport de réduction du système de transmission de mouvement. Pour une liaison prismatique, ce rapport est homogène à l'inverse d'une distance, et pour une liaison rotoïde, il est sans dimension.

$q^i = q_m^i/n^i$ est la coordonnée généralisée.

Si on néglige les élasticités, les jeux, ..., l'équation dynamique de ce système peut s'écrire comme suit:

$$K_{ac}^i \cdot K_c^i \cdot u^i = J^i \cdot \ddot{q}_m^i + \Gamma^i/n^i \quad (I.3)$$

En remplaçant Γ^i par son expression donnée par l'équation (I.1) on obtient:

$$u^i = \frac{n^i}{K_{ac}^i \cdot K_c^i} \left[J^i + \frac{a^{ii}(q)}{(n^i)^2} \right] \cdot \ddot{q}^i + \frac{T^i}{n^i \cdot K_{ac}^i \cdot K_c^i} \quad (I.4)$$

où:

$$T^i = \sum_{j=1, j \neq i}^n a^{ij}(q) \ddot{q}^j + \sum_{j,k=1}^n b^{i,jk}(q) \dot{q}^j \dot{q}^k - G^i(q) + F^{ii} \dot{q}^i + H^{ii} \operatorname{sgn}(\dot{q}^i) \quad (I.5)$$

I-2 - Représentation d'état:

On peut écrire le modèle dynamique sous forme

d'équation d'état de plusieurs façons.

Posons :

I:matrice unité d'ordre n,

O: matrice nulle d'ordre n,

$$L = [O \ I] \quad \text{et} \quad K = \begin{bmatrix} O & I \\ O & O \end{bmatrix}$$

Si on choisit comme vecteur d'état, le vecteur :

$$\dot{x}_p = [q^t \ \dot{q}^t]^t$$

$2n \times 1$

On peut écrire :

$$i. \dot{x}_p = A_p \cdot x_p + B_p \cdot u + f_p$$

$2n \cdot 1 \quad 2n \cdot 2n \quad 2n \cdot 1 \quad 2n \cdot n \quad n \cdot 1 \quad 2n \cdot 1$

avec:

$$A_p = \begin{bmatrix} O & I \\ A^{-1} \cdot G' & -A^{-1} (B' + F) \end{bmatrix} \quad ; \quad B_p = \begin{bmatrix} O \\ A^{-1} \end{bmatrix}$$

$$f_p = \begin{bmatrix} O \\ -A^{-1} \cdot H \cdot \text{Sgn}(\dot{q}) \end{bmatrix} \quad ; \quad u = \Gamma$$

$$ii. \dot{x}_p = \underset{(2n \cdot 2n)}{K} \cdot \underset{(2n \cdot 1)}{x_p} + \underset{(2n \cdot n)}{B_p} \cdot \underset{(n \cdot 1)}{u} + \underset{(2n \cdot 1)}{f'_p} = K \cdot x_p + L^t \cdot A^{-1} \cdot (u - T); \dot{x}_p (2n \cdot 1)$$

(I.7)

avec:

$$f'_p = \begin{bmatrix} O \\ -A^{-1}T \end{bmatrix} \quad \text{et} \quad T = B - G + F \cdot \dot{q} + H \cdot \text{sgn}(\dot{q})$$

Notons également:

$$A_p = \begin{bmatrix} O & I \\ A_{p1} & A_{p2} \end{bmatrix}, \quad B_p = \begin{bmatrix} O \\ B_{p2} \end{bmatrix}, \quad f_p = \begin{bmatrix} O \\ f_{p2} \end{bmatrix}, \quad f'_p = \begin{bmatrix} O \\ f'_{p2} \end{bmatrix}$$

$$\text{donc: } B_{p2} = A^{-1}$$

donc: $B_{p2} = A^{-1}$

De plus, l'équation d'état du modèle de référence est la suivante:

$$\dot{x}_m = A_m \cdot x_m + B_m \cdot u_m \quad (I.8)$$

$2n \cdot 1$ $2n \cdot 2n$ $2n \cdot 1$ $2n \cdot p$ $p \cdot 1$

avec:

$$A_m = \begin{bmatrix} 0 & I \\ A_{m1} & A_{m2} \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ B_{m2} \end{bmatrix}$$

où A_m doit être une matrice d'Hurwitz et u_m est de dimension ($p \times 1$), p étant égal à n , $2n$ ou $3n$, selon que l'on ait des consignes, respectivement de position, de position et de vitesse, ou de position, vitesse et accélération. Des détails sur le choix du modèle de référence seront donnés ultérieurement.

Dans l'espace généralisé, la notion d'écart de configuration du R.M., peut être définie comme suit:

• écart de position:

$$\epsilon_p = q_d - q$$

• écart de vitesse:

$$\epsilon_v = \dot{q}_d - \dot{q}$$

• écart d'accélération:

$$\epsilon_a = \ddot{q}_d - \ddot{q}$$

où $q_d, \dot{q}_d, \ddot{q}_d$ sont respectivement les vecteurs de coordonnées, vitesses, et d'accélérations généralisées désirées.

L'écart d'état ϵ qui représente l'écart entre l'état

désiré x_d et l'état réel du processus x_p :

$$\epsilon = x_d - x_p = [\epsilon_p^t \ \epsilon_v^t]^t$$

On définit aussi la notion d'erreur e , à partir des états du modèle de référence et du processus, comme suit:

$$e = x_m - x_p \quad \text{et} \quad \dot{e} = \dot{x}_m - \dot{x}_p$$

Donc: $e = [e_p^t \ e_v^t]^t$

avec: $e_p = q_m - q \quad \text{et} \quad e_v = \dot{q}_m - \dot{q}$

de plus on posera :

$$e_a = \ddot{q}_m - \ddot{q}$$

Ceci étant dans l'espace généralisé, il sera nécessaire de passer d'un espace à l'autre; car d'une part, la tâche à effectuer est définie dans l'espace opérationnel dans lequel le problème de la commande peut être divisé en deux sous-problèmes qui sont la commande de la position et la commande de l'orientation de l'organe terminal du R.M.; d'autre part, l'évolution de l'organe terminal est le résultat de la variation de la configuration du R.M.

A partir de ces considérations, apparaît la nécessité de disposer de certaines transformations qui autorisent le passage de l'espace opérationnel à l'espace généralisé et vice versa. Ces transformations sont appelés: modèles inverses dans le premier cas et directs dans le second. On distingue:

1. le modèle géométrique direct qui est la fonction qui permet le calcul de la situation x , de l'organe terminal à partir de la configuration q , du R.M. :

$$x = f(q, \theta_s)$$

θ_s est un vecteur dont les éléments sont les paramètres

structuraux du R.M.

2. le modèle géométrique inverse est la fonction qui permet le calcul de la configuration q du R.M., à partir de la connaissance de la situation x , de l'organe terminal :

$$q = g(x, \theta_s)$$

3. Le modèle différentiel direct est la fonction qui permet le calcul de la différentielle de la situation dx , à partir de la connaissance de la différentielle de la configuration dq , et de la matrice jacobienne $J(q, \theta_s)$ du R.M.:

$$dx = J(q, \theta_s) \cdot dq$$

4. le modèle différentiel inverse est la fonction qui permet le calcul de la différentielle de situation dq , à partir, d'une part de la connaissance de la différentielle de situation dx , et d'autre part d'une inverse généralisée $G(q, \theta_s)$ de $J(q, x, \theta_s)$ du R.M.:

$$dq = G(q, \theta_s) \cdot dx$$

La matrice de passage homogène ¹³ élémentaire $T_{1,i+1}$ entre le repère lié au corps C_{i-1} et le repère lié au corps C_i définit la situation du corps C_i par rapport à son antécédant C_{i-1} . La matrice de passage homogène est le produit des matrices de passage élémentaires $T_{1,i+1}$. Donc la matrice de passage homogène, $T_{1,n}$ entre le repère lié à l'organe terminal et celui lié à la base, a pour expression:

$$T_{1,n+1} = T_{1,2} \times \cdots \times T_{1,i+1} \times \cdots \times T_{n,n+1}$$

Chapitre II

CHAPITRE II

COMMANDE ADAPTATIVE A MODELE DE REFERENCE DES ROBOTS MANIPULATEURS

II-1- introduction:

La commande adaptative semble être ancienne, et même plus ancienne que la commande par retour d'état, seulement les chercheurs ne s'en sont intéressés qu' après 1950.

Parmi les différents types de commande adaptative, le type à modèle de référence (fig.II.1) connue sous le nom de MRAS (:model reference adaptive system) est l'un des plus importants' grâce à sa facilité d'implantation et à l'importance de sa vitesse d'adaptation. Dans ce type de commande, le modèle de référence est excité par les mêmes entrées que le processus. En termes d'entrées et états du modèle, celui-ci spécifie un critère de performances donné. Dans ce cas la comparaison entre critère désiré, et critère mesuré est obtenue directement par la comparaison des sorties ou états du système de commande adaptatif, à ceux du modèle de référence. La différence est utilisée par le mécanisme d'adaptation, soit pour générer un signal auxiliaire à l'entrée, pour minimiser cette différence, soit pour modifier les paramètres du système d'adaptation.

La vitesse d'adaptation est l'un des avantages les plus importants. Cette vitesse est due à l'obtention immédiate de la différence entre les deux critères à partir de la comparaison entre les états du modèle et ceux du

systeme. Parcontre il faut avoir à priori, des informations sur la structure (parfois du modèle, et parfois du systeme lui même) afin de pouvoir implanter ce type de commande.

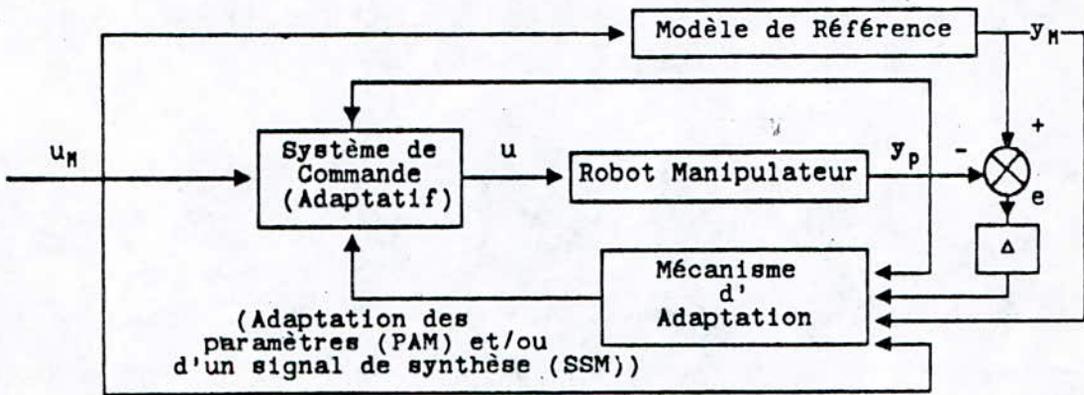


fig.II.1. Shéma de principe de MRAS(parallèle).

Les états du processus et du modèle sont supposés observables.

Le schéma ci-dessus, représente la commande de type MRAS à configuration parallèle; or en réalité il y a plusieurs types de MRAS. Selon la structure on distingue[13]:

- MRAS parallèle,
- MRAS série-parallèle,
- MRAS série.

Mais la structure la plus utilisée est celle donnée

par le schéma ci-dessus (i.e. parallèle).

Il est à noter qu'il ne faut pas confondre le critère de performances désiré, spécifié par le modèle, au critère qui sert uniquement pour le mécanisme d'adaptation.

Trois approches de base sont connues dans le domaine de MRAS. Elles diffèrent par la théorie sur laquelle elles sont basées. Ces trois approches sont les suivantes:

- celle qui est basée sur l'optimisation locale des paramètres,
- celle qui est basée sur la stabilisation à l'aide de la théorie de Lyapunov.
- et celle basée sur la stabilisation à l'aide de la théorie d'hyperstabilité de Popov.

II-2- MRAS basée sur l'optimisation locale des paramètres:

C'est la première méthode qui a été utilisée dans le domaine de MRAS. Utilisée initialement par S.Dubowsky et D.T. Desforges sans équations d'état en 1979, puis par S.Dubowsky et R.Kornbluth en 1984 avec équations d'état. L'idée de base est la suivante:

- faire varier les gains et donc les paramètres du système de commande, pour minimiser un critère quadratique exprimant l'écart entre le modèle de référence et le système réel (i.e. annuler l'écart $e = x_m - x_p$).

Ces méthodes supposent que:

- au début on démarre avec un petit écart c'est à dire :
 $A_m - A_p$ et $B_m - B_p$ sont petits,

• la vitesse de variation des paramètres est faible.

Pour ce type de commande on distingue trois algorithmes de base qui sont basés sur l'une des méthodes suivantes:

- méthode du gradient,
- méthode du gradient conjugué,
- et méthode de la plus grande pente.

Elles utilisent les fonctions de sensibilité. Le schéma ci-dessous correspond à la méthode de la plus grande pente:

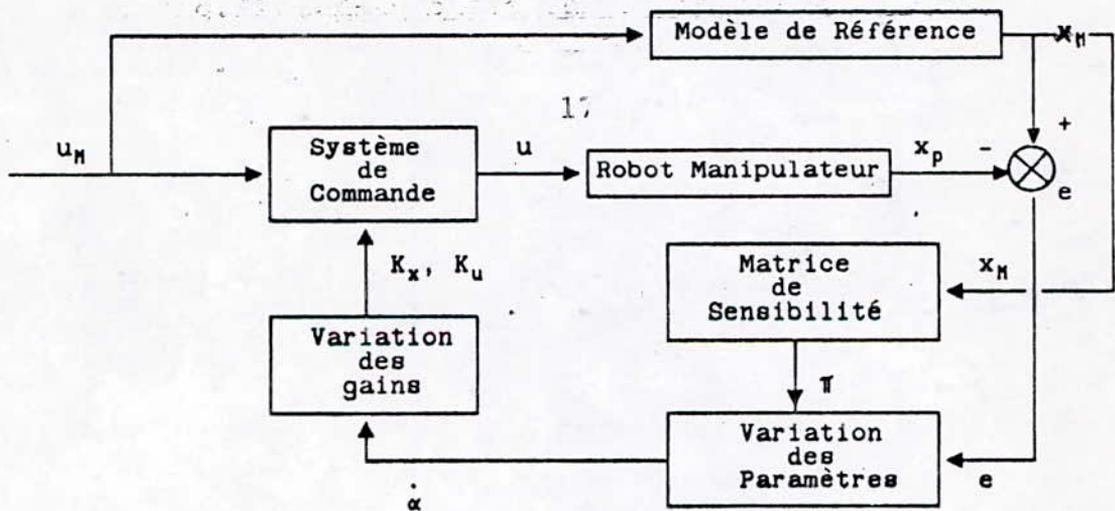


fig.II.2. MRAS basée sur l'optimisation locale des paramètres

La dynamique du système est considérée de la forme:

$$\dot{x}_p = f(x_p, \theta_p, u_m)$$

Celle du modèle de référence est donnée par:

$$\dot{x}_m = f(x_m, \theta_m, u_m)$$

où θ est le vecteur des paramètres.

Soit $e = x_m - x_p$

$$\text{et } V(e) = (e^t P e) / 2$$

où P est une matrice définie positive.

Si on choisit :

$$\dot{\alpha} = -K \cdot \frac{\partial V}{\partial \theta_p} = -K \cdot \frac{\partial e^t}{\partial \theta_p} \cdot Q \cdot e \quad (\text{II.1})$$

où: K est un scalaire positif.

à partir de:

$$\frac{\partial e}{\partial \theta_p} = -\frac{\partial x_p}{\partial \theta_p} = \frac{\partial x_m}{\partial \theta_m} = \pi \quad (\text{II.2})$$

π étant la matrice de sensibilité; et d'après l'équation (II.1) on obtient la loi d'adaptation suivante:

$$\dot{\alpha} = -K \pi^t \cdot Q \cdot e \quad (\text{II.3})$$

avec:

$$\pi = \frac{\partial f_m}{\partial x_m} \pi + \frac{\partial f_m}{\partial \theta_m} \quad (\text{II.4})$$

L'inconvénient de cette méthode est qu'elle ne s'intéresse pas de la stabilité du système à commander. Les méthodes qui suivent s'en intéressent.

II-3- MRAS basée sur la stabilisation à l'aide de la théorie de Lapunov:

Cette méthode a été utilisée par H. Durrant-Whyte, K.Y. Lim et M. Eslami, N. Sadegh et R. Horowitz, D.T. Gavel et T.C. Hsia, et H. Seraji.

Le schéma de cette approche est le suivant:

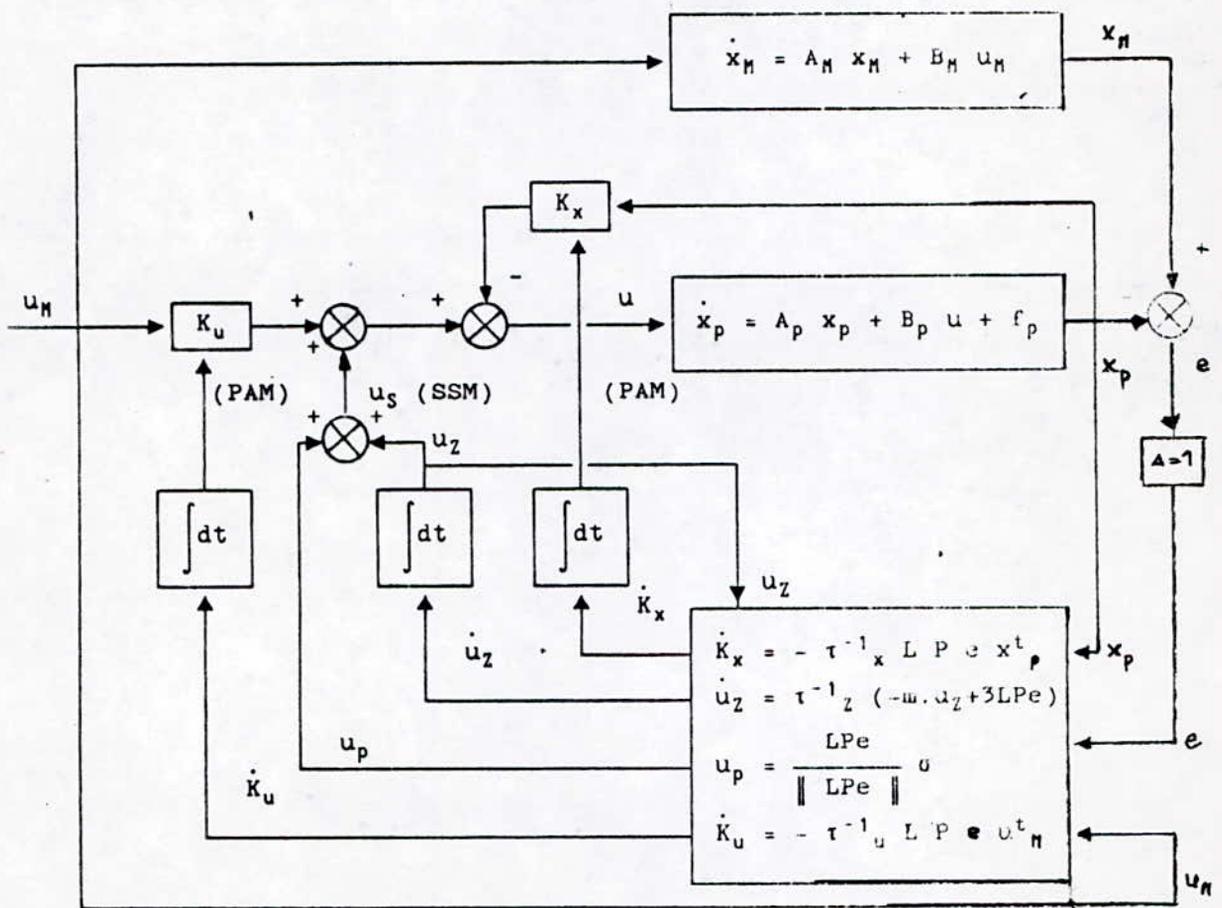


fig.II.3. MRAS à stabilisation à l'aide de la théorie de Lyapunov.

Le principe de cette méthode est d'adapter les paramètres K_x , et K_u (PAM: parameter adaptation method) et engendrer un signal de synthèse (SSM: signal synthesis method).

Le signal de synthèse u_s , a pour rôle d'améliorer la vitesse de convergence de l'erreur e vers zéro, et d'atténuer l'effet de perturbation du terme fp du processus.

Si l'erreur est e , son équation d'état est :

$$\dot{e} = \dot{x}_m - \dot{x}_p = A_m \cdot e + (A_m - A_s) \cdot x_p + (B_m - B_s) \cdot u_m - fp + B_p \cdot u_s \text{ (II.5)}$$

avec:

$$\begin{aligned} \dot{x}_m &= A_m \cdot x_m + B_m \cdot u_m \\ \dot{x}_p &= A_p \cdot x_p + B_p \cdot u + fp \end{aligned}$$

et où:

$$\begin{aligned} A_s &= A_p - B_p \cdot K_x \\ B_s &= B_p \cdot K_u \end{aligned}$$

A_s et B_s sont les matrices de l'équation d'état du processus doté du gain de contre-réaction K_x , du gain d'anticipation K_u et commandé par le signal u_m :

$$\dot{x}_p = A_s \cdot x_p + B_s \cdot u_m + fp$$

si on pose:

$$\begin{aligned} A_m - A_p &= -B \cdot \dot{K}_x \\ B_m - B_p &= B_p \cdot \dot{K}_u \end{aligned}$$

on aura:

$$\dot{e} = A_m \cdot e + B_p \cdot [K - \dot{K}_x; \dot{K}_u - K_u] \begin{bmatrix} x_p \\ u_m \end{bmatrix} - fp - B_p \cdot u_s$$

La méthode suppose que la vitesse d'évolution du processus est faible devant la vitesse d'adaptation des paramètres et d'élaboration du signal de synthèse. Donc on peut supposer A_p et B_p constantes, c'est à dire:

$$\dot{K}_x = 0 \text{ et } \dot{K}_u = 0$$

Si en premier lieu, on suppose que $f_p = 0$ (i.e. $u_p = 0$), la méthode consistera donc à adapter K_x et K_u , et à engendrer le signal $u_z = u_z$ de telle sorte que V soit une fonction de Lyapunov, avec:

$$V = e^t P e + \tau_x \cdot \text{tr}(K_x - \dot{K}_x)^t \cdot B_{p2} \cdot (K_x - \dot{K}_x) + \tau_u \cdot \text{tr}(K_u - \dot{K}_u)^t \cdot B_{p2} (K_u - \dot{K}_u) + \tau_z \cdot \sum_{i=1}^3 u^t \cdot z_i \cdot B_{p2} \cdot u_{zi}$$

$$\text{où:} \quad (II.6)$$

$$u_z = \sum_{i=1}^3 u_{zi}$$

avec:
 $P, \tau_x, \tau_u \text{ et } \tau_z$

matrices symétriques, définies positives.

et puisque $B_{p2} = A^{-1}$ alors B_{p2} l'est aussi.

Le mécanisme d'adaptation étant choisi tel que:

$$\begin{aligned} \dot{K}_x &= -\tau_x^{-1} \cdot L.P.e \cdot x_p^t \\ \dot{K}_u &= \tau_u^{-1} \cdot L.P.e \cdot u_m^t \\ \dot{u}_{zi} &= \tau_z^{-1} \cdot (-m \cdot u_{zi} + L.P.e) \end{aligned} \quad (II.7)$$

soit:

$$\dot{u}_z = \tau_z^{-1} \cdot (-m \cdot u_z + 3 \cdot L.P.e)$$

et en dérivant V donné par l'expression (II.5) on obtient:

$$\dot{V} = -e^t \cdot Q \cdot e - 2 \cdot m \cdot \sum_{i=1}^3 u_{zi}^t \cdot B_{p2} \cdot u_{zi} \quad (II.8)$$

avec:

$$A_m^t \cdot P + P \cdot A = -Q \quad (II.9)$$

et A_m étant une matrice d'Hurwitz.

m doit être tel que:

$$m > - \frac{\mu_{\min}(Q) \cdot ||e||^2}{2 \cdot (\mu_{\min}(A^{-1})) \cdot ||u_z||^2} \quad (II.10)$$

pour assurer la convergence de l'erreur e vers zéro.

avec:

$\mu_{\min}(Q)$ représente la plus petite des valeurs propres de la matrice Q , et $||mat||$ la norme euclidienne de la matrice mat .

m permet également de faire varier le rapport $\phi = -\dot{V}/V$ qui mesure la vitesse de convergence.

Si maintenant on suppose f_{p2} non nul, il est nécessaire de rajouter un signal de synthèse supplémentaire u_p de telle manière que $u_s = u_z + u_p$.

u_p est tel que:

$$u_p = \frac{L.P.e}{||L.P.e||} \cdot \sigma \quad (II.11)$$

où σ dépend en particulier de $||f_{p2}||_{\max}$.

II-4- MRAS basée sur la stabilisation à l'aide de la théorie de Popov:

Cette approche a été utilisée pour la première fois dans ce domaine par R. Horowitz et M. Tomizuka (1980) et reprise (1986).

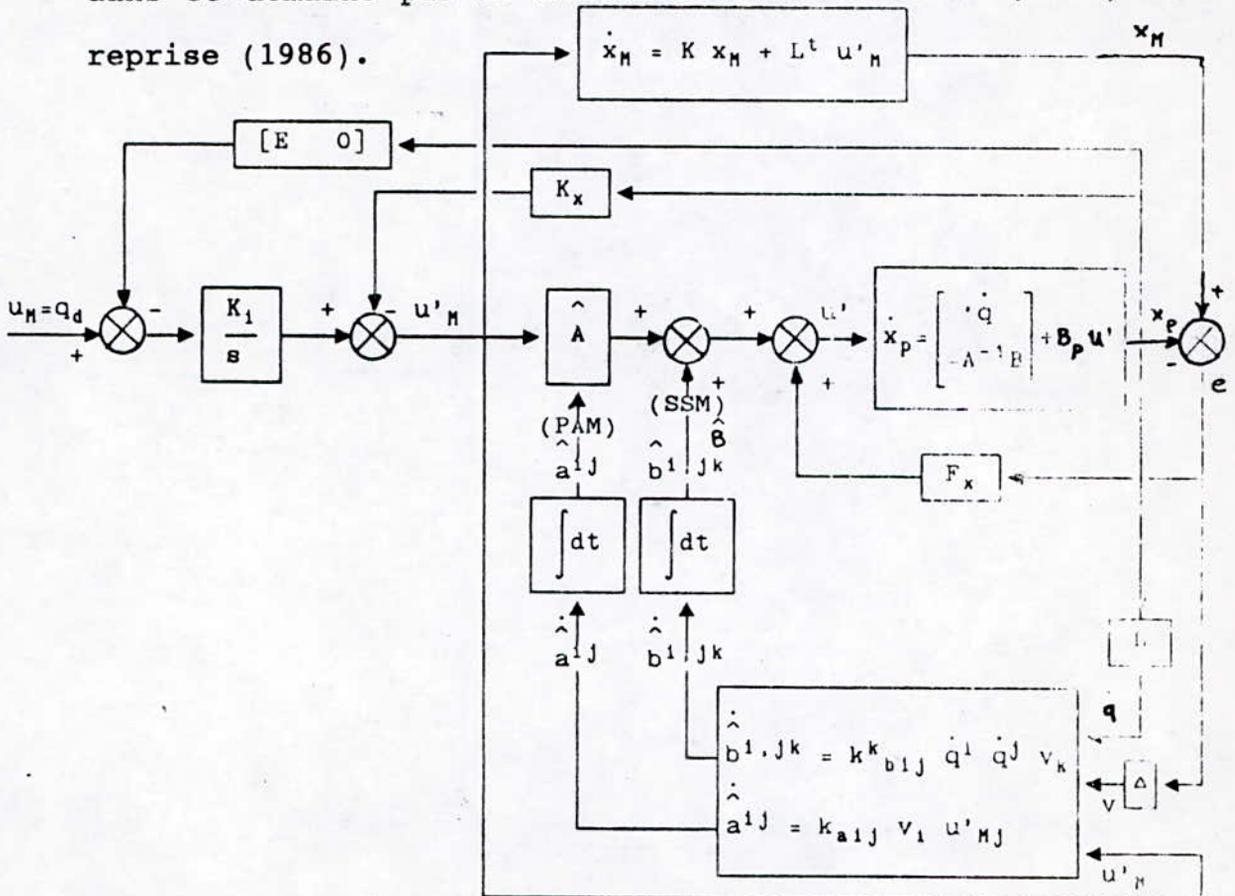


fig.II.4. MRAS à stabilisation à l'aide de l'hyperstabilité de Popov

Dans cette méthode, ce sont directement les éléments a^{ij} et $b^{i,jk}$ du modèle dynamique qui sont adaptés, et non pas les gains de contre-réaction et d'anticipation du système de commande.

on a :

$$\begin{aligned} \frac{d}{dt} \hat{a}^{ij} &= K_{a^{ij}} \cdot v_i \cdot \dot{u}_{mj} & K_{a^{ij}} > 0 \\ \frac{d}{dt} \hat{b}^{i,jk} &= K^{k_{bij}} \cdot \dot{q}^i \cdot \dot{q}^j \cdot v_k & K^{k_{bij}} > 0 \end{aligned}$$

avec :

$$v = \Delta \cdot e$$

Soient : $Fx = [F_p \ F_v]$, $Kx = [K_p \ K_v]$ et $\Delta = [\Delta_p \ \Delta_v]$

A partir des équations d'état du modèle et du processus :

$$\dot{x}_m = K \cdot x_m + L^r \cdot u'_m$$

si on pose : $u'_m = \ddot{q}_m$

$$\dot{x}_p = \begin{bmatrix} \dot{q} \\ -A^{-1} \cdot B \end{bmatrix} + B_p \cdot u'$$

et sachant que : $u' = Fx \cdot e + B' + A' u'_m$

l'équation d'état de l'erreur est :

$$\dot{e} = \begin{bmatrix} 0 & I \\ -A^{-1} \cdot F_p & -A^{-1} \cdot F_v \end{bmatrix} \cdot e + B_p(-w) = A_e \cdot e + B_p(-w)$$

avec :

$$w = (A' - A) \cdot u'_m + (B' - B)$$

Par conséquent, la fonction de transfert $G(s) = \Delta (s \cdot E - A_e)^{-1} \cdot B_p$

avec :

$$G(s) = -V(s)/W(s)$$

Les auteurs démontrent à l'aide de la théorie d'hyperstabilité de Popov[2], que si $G(s)$ est réelle, strictement positive, l'erreur e est asymptotiquement stable.

De plus les gains sont choisis de telle sorte que les polynômes matriciels:

$$s^3.E + s^2.Kv + s.Kp \quad \text{et} \quad s^2.E + s.Kv + Kp$$

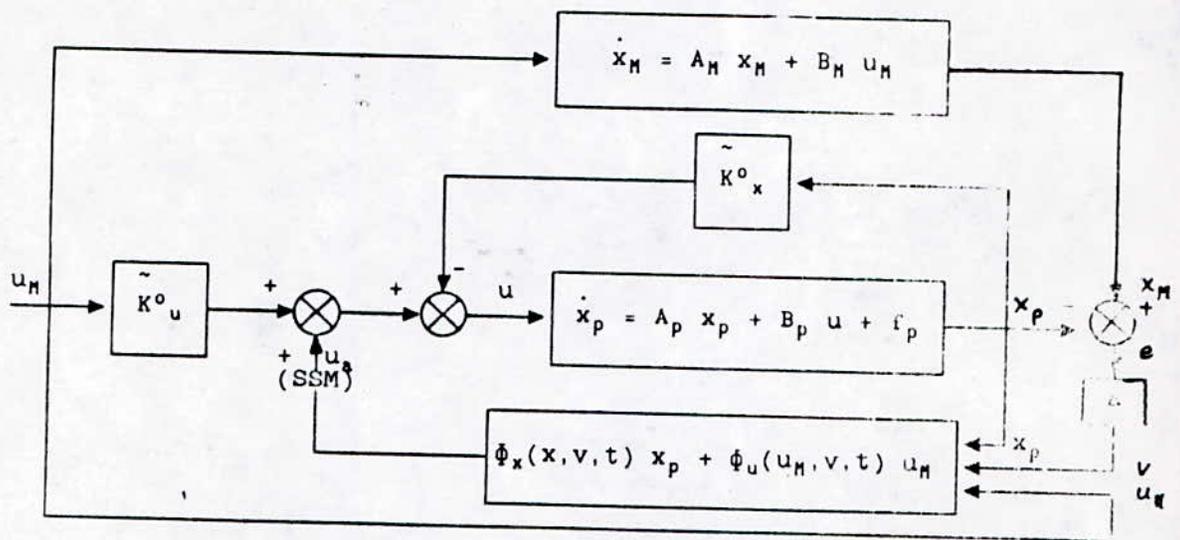
soient asymptotiquement stables.

Si le système est idéal (i.e. $u'_m = \ddot{q}$), la fonction de transfert est:

$$\frac{q(s)}{q_d(s)} = \frac{Ki}{ki + s \cdot Kp + s^2 \cdot Kv + s^3}$$

Il y a aussi une autre approche pour cette méthode. Celle-ci a été présentée par A. Balestrino, G. De Maria et L. Sciavicco en 1981, puis en 1985; D.P. Stoten en 1983; S. Nicosia et P. Domei en 1984; M. Vukobratovik et N. Kircanski en 1984; S.N. Singh en 1985; J. Katende, A.J. Alos et B.O. Amobamise en 1985; Y.F. Zeng et H. Hemami en 1985; H.R. Asare et D.G. Wilson en 1987.

Ces méthodes consistent à imposer des gains constants de contre-réaction et d'anticipation au système de commande et à engendrer un signal de synthèse (ssm). Son schéma de principe est le suivant:



soient:

$$K0x' = A0.L.(A_m - A0p)$$

$$K0u' = A0.L.B_m$$

$$u = -K0x'.x_p + K0u'.u_m + \Phi_x.x_p + \Phi_u.u_m$$

où $A0$ est une matrice constante correspondant à une valeur nominale de A .

on a:
$$e = x_m - x_p$$

donc:
$$\dot{e} = A_m.e + B_p.[A.L.(A_m - A_p) + A0.L.(A_m - A0p) - \Phi_x].x_p$$

$$+ B_p.[(A - A0).L.B^m - \Phi_u].u_m - f_p \quad \circ u$$

bien
$$\dot{e} = A_m.e - L^t.w$$

où
$$w = B_{p2}.(\Phi_p - h_p).q + B_{p2}.(\Phi_v - h_v).\dot{q} + B_{p2}.(\Phi_u - h_u).u_m$$

$$+f_{p2} \text{ avec: } h_p = A \cdot (A_{m1} - A_{p1}) + A0 \cdot (A_{m1} - A0_{p1})$$

$$h_v = A \cdot (A_{m2} - A_{p2}) + A0 \cdot (A_{m2} - A0_{p2})$$

$$h_u = (A - A0) \cdot B_{m2}$$

$$\text{soit } w = B_{p2} \cdot (\phi x - h_x) \cdot x_p + B_{p2} \cdot (\phi u - h_u) \cdot u_m + f_{p2}$$

$$\text{avec: } h_x = (h_{pt} \ h_{vt})^t$$

$$\text{suposons } f_{p2} = 0$$

$$\text{si } \Delta = L \cdot P \text{ alors } G(s) = \Delta \cdot (s \cdot I - A_e)^{-1} \cdot B_p$$

est réelle positive strictement[].

Les auteurs montrent que dans ce cas l'erreur est asymptotiquement stable. Celle-ci fait l'objet de la première condition de Popov.

P est la matrice définie pour le critère de Lyapunov ($P > 0$) à l'aide de l'équation (II.9). Ainsi, la fonction de transfert: $G(s) = L \cdot P \cdot (s \cdot I - A_m)^{-1} \cdot L^t = P_{22} \cdot (s \cdot I - A_m)^{-1}$ est positive car :

$$P_{22} \text{ et } (s \cdot I - A_m)^{-1}$$

le sont si A_m vérifie la condition d'Hurwitz .

La deuxième condition de Popov est []:

$$\int_0^t v^t \cdot w \cdot d\tau \geq -\mu_0^2 \quad (\text{II.12})$$

$$\text{avec: } v = \Delta \cdot e$$

cette condition est vérifiée si (avec $\mu_0 = 0$) on choisit:

$$\phi x - Kx \cdot \frac{v}{|v|} \cdot (\text{sgn} x_p)^t$$

$$\phi u - Ku \cdot \frac{v}{|v|} \cdot (\text{sgn} u_m)^t \quad (\text{II.13})$$

avec:

$$\text{sgn} x_p = [\text{sgn} x_{p1} \dots \text{sgn} x_{pn}]^t$$

Kx et Ku doivent vérifier :

$$\begin{aligned} Kx &\geq \frac{\max ||R||}{\mu_{\min, \min}(B_{p2})} \\ Ku &\geq \frac{\max ||S||}{\mu_{\min, \min}(B_{p2})} \end{aligned} \quad (\text{II.15})$$

avec: $R = B_{p2} \cdot h_x$ et $S = B_{p2} \cdot h_u$

où $\mu_{\min, \min}(\text{mat})$ est le minimum de la plus petite des valeurs propres de la matrice mat, au cours du mouvement.

Chapitre III

CHAPITRE III

SIMULATION

III-1- Introduction:

On connaît deux grandes classes de simulation [4]: numérique et en temps réel. Chacune se ramifie en plusieurs méthodes. La deuxième comporte la simulation analogique et la simulation digitale; alors que la première connaît deux méthodes principales qui sont: le calcul de la réponse indicielle et la résolution des des équations d'état. Cette dernière peut se faire selon trois méthodes connues:

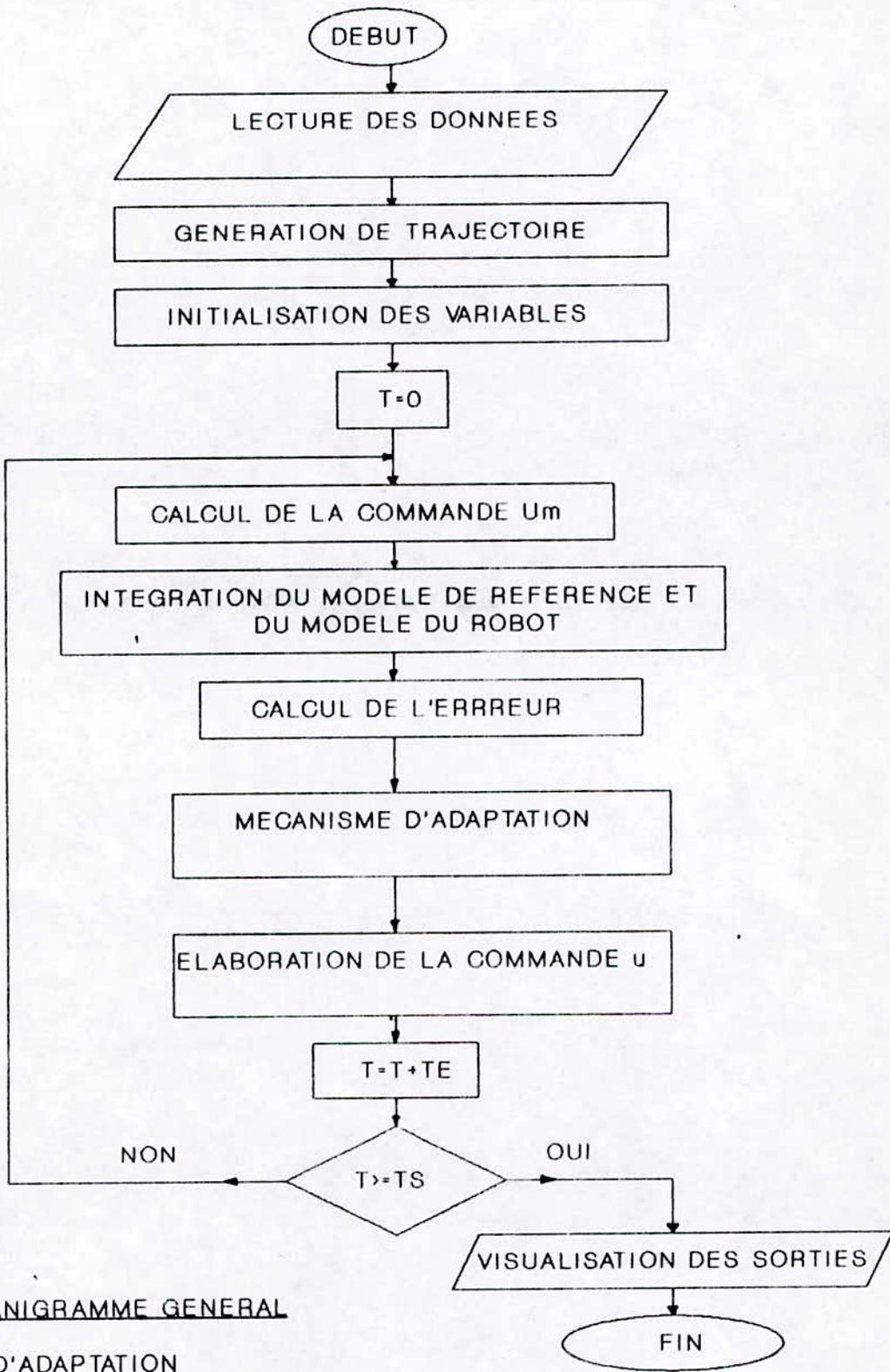
- méthode d'Euler,
- méthode de Runge-Kutta,
- et méthode de la matrice transition d'état.

Pour notre cas (simulation numérique) on choisit la méthode de résolution des équations d'état car le système (R.M.) est multivariable. La méthode d'intégration pour le modèle de simulation du R.M. est celle de Runge-Kutta (d'ordre quatre), qui convient même pour les systèmes non-linéaires. On l'utilise aussi pour intégrer le modèle de référence.

Le pas d'intégration est nettement plus petit que la période d'échantillonnage (d'adaptation).

La procédure générale d'adaptation est la suivante:

- 1- generer la trajectoire désirée x_d (pour les détails, voir paragraphe III.7),



- 2- calculer la commande u_m (pour avoir $x_m = x_d$),
- 3- intégrer le modèle de référence et le modèle de simulation du robot considéré, et ceci sur une période d'échantillonnage, et en déduire l'erreur,
- 4- activer le mécanisme d'adaptation(l'algorithme), en utilisant l'erreur et d'autres informations sur le système(R.M.),
- 5- élaborer la loi de commande u ,
- 6- incrémenter le temps, s'il est égal au temps final (donné par le dernier triplet de la trajectoire désirée) alors visualiser les résultats (courbes) et fin, sinon retourner à 2.

(voir algorithme correspondant:fig.III.1).

III-2- Méthodes d'intégration:

Pour un système quelconque décrit par l'équation :

$$\dot{x} = f(x, u)$$

l'emploi de la méthode de Runge-Kutta d'ordre 4 nécessite l'algorithme suivant [4]:

$$\begin{aligned} f' &= f(x_k, u_k) \\ x_{k+1/2}' &= x_k + f' \cdot \Delta t / 2 \\ f'' &= f(x_{k+1/2}', u_{k+1/2}') \\ x_{k+1/2}'' &= x_k + f'' \cdot \Delta t / 2 \\ f''' &= f(x_{k+1/2}'', u_{k+1/2}'') \\ x_{k+1}''' &= x_k + f''' \cdot \delta t \\ f^{(4)} &= f(x_{k+1}''', u_{k+1}''') \end{aligned}$$

et:

$$x_{k+1} = x_k + (f' + 2f'' + 2f''' + f^{(4)}) \cdot \Delta t / 6$$

Pour un système linéaire l'équation d'état est :

$$\dot{x} = A \cdot x + B \cdot u$$

si on suppose que u_k est constant pendant un pas d'intégration, on obtient:

$$x_{k+1} = P_r \cdot x_k + Q_r \cdot B \cdot u_k$$

avec:

$$P_r = I + A \cdot \Delta t + A^2 \cdot \frac{\Delta t^2}{2} + A^3 \cdot \frac{\Delta t^3}{6} + A^4 \cdot \frac{\Delta t^4}{24}$$

$$Q_r = \Delta t \cdot \left(I + A \cdot \frac{\Delta t}{2} + A^2 \cdot \frac{\Delta t^2}{6} + A^3 \cdot \frac{\Delta t^3}{24} \right)$$

Remarque: Cette méthode est utilisée uniquement pour intégrer le modèle de référence.

Pour intégrer le processus, on écrit le modèle dynamique sous la forme :

$$\dot{x} = [q \quad \dot{q}]^T = h(k, x)$$

avec d'après l'équation (I.3) donnant Γ :

$$\ddot{q} = A^{-1} \cdot (\Gamma - B(q, \dot{q}) + G(q) - F \cdot \dot{q} - H \cdot \text{sgn}(\dot{q}))$$

d'après la méthode de Runge-Kutta:

$$x(k+1) = x(k) + (t_1 + 2 \cdot t_2 + 2 \cdot t_3 + t_4) \cdot t_e / 6$$

avec: $t_1 = h(k, x)$

$$t_2 = h(k + t_e/2, x + t_1/2)$$

$$t_3 = h(k + t_e/2, x + t_2/2)$$

$$t_4 = h(k + t_e, x + t_3)$$

k est l'instant d'échantillonnage et t_e le pas de calcul (=période d'échantillonnage).

t_1, t_2, t_3, t_4, x et h étant des vecteurs de dimension $2n$.

III-3- Exemple du double pendule:

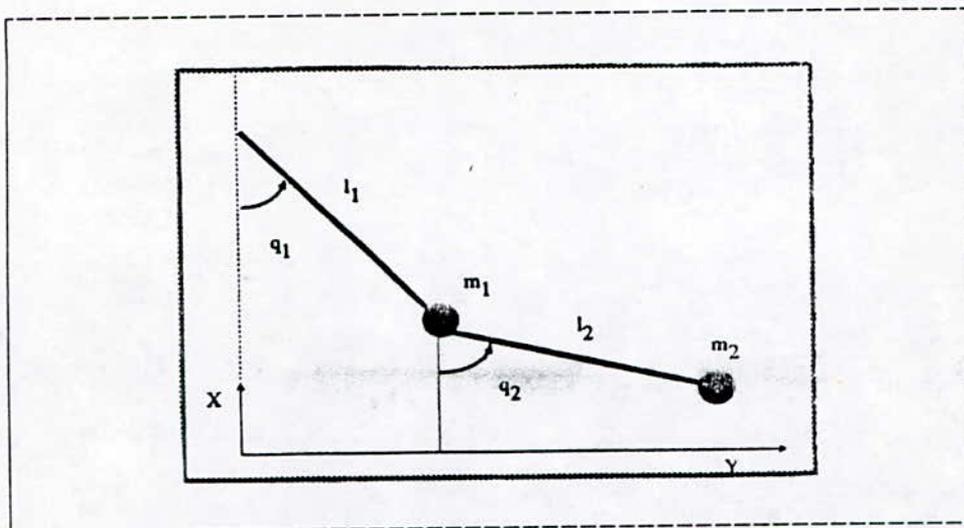


fig.III.2. Structure du robot double-pendule.

Les équations du mouvement du robot double pendule sont:

$$\begin{aligned}
 \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} &= \begin{bmatrix} (m_1+m_2)(l_1)^2 & m_2 \cdot l_1 \cdot l_2 \cdot \cos\alpha \\ m_2 \cdot l_1 \cdot l_2 \cdot \cos\alpha & m_2 \cdot (l_2)^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \\
 &+ \begin{bmatrix} 0 & m_2 \cdot l_1 \cdot l_2 \cdot \sin\alpha \\ m_2 \cdot l_1 \cdot l_2 \cdot \sin\alpha & 0 \end{bmatrix} \begin{bmatrix} (\dot{q}_1)^2 \\ (\dot{q}_2)^2 \end{bmatrix} + \\
 &+ \begin{bmatrix} (m_1+m_2) \cdot g \cdot l_1 \cdot \sin(q_1) \\ m_2 \cdot g \cdot l_2 \cdot \sin(q_2) \end{bmatrix}
 \end{aligned}$$

q_1 et q_2 sont les position angulaires des liaisons 1 et 2.

$$\alpha = q_1 - q_2$$

Γ_1 et Γ_2 désignent les couples exercés sur chaque liaison rotoïde, et g est l'accélération de pesanteur.

Les deux équations comportent des termes non linéaires et fortement couplés. Pour intégrer ce modèle à l'aide de

la méthode de Runge-Kutta on tire les expressions de \ddot{q}_1 et \ddot{q}_2 , après calcul on obtient:

$$\ddot{q}_1 = \frac{r_1 r_2 \cdot l_1 \cdot \cos \alpha / l_2 (m_1 + m_2) \cdot g \cdot l_1 \cdot \sin q_1 + m_2 \cdot l_1 (g \cdot \cos \alpha \cdot \sin q_2 + l_1 \cdot \sin \alpha \cdot \cos \alpha \cdot (\dot{q}_1)^2 - l_2 \cdot \sin \alpha (\dot{q}_2)^2)}{(m_1 + m_2) \cdot (l_1)^2 - m_2 (l_1)^2 \cdot \cos^2 \alpha} ,$$

$\ddot{q}_2 = \frac{(r_1 - m_2 \cdot l_1 \cdot l_2 \cdot \sin \alpha \cdot (\dot{q}_2)^2 - (m_1 + m_2) l_1 (g \cdot \sin q_1 + l_1 \ddot{q}_1))}{m_2 \cdot l_1 \cdot l_2 \cdot \cos \alpha}$; pour pouvoir appliquer l'intégration de Runge-kutta on écrit:

$$x = [q^t \dot{q}^t]^t \text{ avec: } q^t = [q_1 \ q_2] \text{ et } \dot{q}^t = [\dot{q}_1 \ \dot{q}_2]$$

et on intègre x pour avoir les coordonnées généralisées.

Remarque:

Nous avons validé le modèle de simulation du double pendule en introduisant des valeurs de r_1 et r_2 pour lesquelles nous avons calculé q_1 et q_2 (manuellement), puis appliqué ces mêmes valeurs de commande au modèle; les résultats sont satisfaisants.

III-4- Cas du robot TH8 à 3 liaisons:

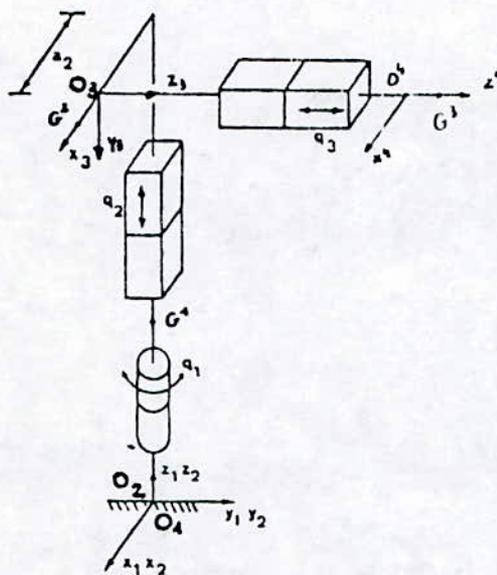


fig.III.3. Structure du robot TH8 à 3 liaisons.

Ce robot est obtenu à partir du robot TH8 à six liaisons, en figeant les trois dernières liaisons. Le nouveau troisième corps est doté des masses des 3°, 4°, 5° et 6° corps. C'est un robot de complexité moyenne.

Suivant les axes placés sur le schéma on a[11]:

$$\begin{aligned}\Gamma^1 &= [\hat{c}^1 + \hat{b}^2 + \hat{b}^3 + (a^2 + \bar{x}^2)^2 m^2 + (a^2)^2 + q^3 + \bar{z}^3)^2 m^3] \ddot{q}^1 + \\ &\quad + a^2 m^3 \ddot{q}^3 + 2(q^3 + \bar{z}^3) m^3 \dot{q}^1 \dot{q}^3 \\ \Gamma^2 &= (m^2 + m^3) (\ddot{q}^2 + g) \\ \Gamma^3 &= m^3 \ddot{q}^3 + m^3 a^2 \ddot{q}^1 - m^3 (q^3 + \bar{z}^3) (\dot{q}^1)^2\end{aligned}$$

où seulement les exposants qui suivent une parenthèse fermée, sont des exposants.

Γ^i $i=1,2,3$ est l'effort généralisé à la liaison i .

\hat{c}^1 : inertie du corps c^1 en son centre de gravité G^1 , autour d'un axe parallèle à l'axe z^1 ,

b^2 : inertie du corps c^2 en son centre de gravité G^2 , autour d'un axe parallèle à l'axe y^4 ,

b^3 : inertie du corps $c=c^3 + c^4 + c^5 + c^6$ au centre de gravité de c , autour d'un axe parallèle à y^4

a^2 est le paramètre de Denavit-Hartenberg exprimant la distance du point O^2 au point O^3 , projetée sur l'axe x^2 .

x^2 : distance du point O^3 au centre de gravité G^3 du corps c , projetée sur l'axe z^4 .

m^2 : masse de c^2 ,

m^3 : masse de c ,

q^3 : coordonnée généralisée exprimant la distance du point O^3 au point O^4 projetée sur l'axe z^3 ,

\dot{q}^1, \dot{q}^3 : vitesses généralisées associés aux liaisons 1 et 3,

$\ddot{q}^1, \ddot{q}^2, \ddot{q}^3$: accélérations généralisées associées aux liaisons 1, 2, 3 ; g : accélération de la pesanteur.

III-5- Choix du modèle de référence:

Le choix du modèle de référence est une tâche très importante dans une adaptation MRAS car il reflète le comportement désiré pour le système à adapter. Pour avoir à partir d'un système non linéaire découplé, le comportement d'un système linéaire découplé, on choisit un modèle de référence représenté par des équations différentielles du second ordre, linéaires et invariantes. Donc pour une liaison i , l'équation différentielle correspondante est:

$$\ddot{q}_{mi} + a_{1i} \dot{q}_{mi} + a_{0i} q_{mi} = u_{mi} \quad i=1, \dots, n$$

l'équation d'état du modèle de référence étant:

$$\dot{x} = A_m x_m + B_m u_m$$

avec:

$$x_m = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$

les matrices A_m et B_m ont la forme:

$$A_m = \begin{bmatrix} 0 & I \\ A_{m1} & A_{m2} \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ B_{m2} \end{bmatrix}$$

avec:

$$A_{m1} = -\text{diag}(a_{0i}), \quad A_{m2} = -\text{diag}(a_{1i}) \quad \text{et} \quad a_{0i} > 0, \quad a_{1i} > 0$$

$$B_{m2} = I$$

d'autre par on sait que l'équation différentielle, d'un système du second ordre en fonction de ses paramètres est la suivante:

$$u_{mi} = \frac{1}{\omega_{ni}^2} \ddot{q}_{mi} + \frac{2\xi}{\omega_{ni}} \dot{q}_{mi} + q_{mi} \quad i=1, \dots, n$$

et en identifiant à l'équation différentielle plus haut on obtient:

$$a_{1i} = 2 \frac{\xi}{\omega_{ni}} \quad \text{et} \quad a_{0i} = 1$$

III-6- Génération de la trajectoire:

La trajectoire de consigne est définie par un ensemble de N_t triplets $(q, \dot{q}, t)_k$ $k=1, \dots, N_t$. Alors que la trajectoire que doit effectuer le R.M. et qui représente la consigne réelle, est définie par une suite de valeurs des coordonnées, des vitesses et des accélérations généralisées. Ces valeurs sont obtenues par une interpolation polynomiale d'ordre 3, entre deux consignes successives représentées par deux triplets $(q, \dot{q}, t)_k$ et $(q, \dot{q}, t)_{k+1}$. Pour chaque liaison i , cette trajectoire est donnée par les expressions suivantes:

- Coordonnée généralisée : C'est une fonction cubique du temps:

$$q_i = a_{ik} \frac{(t-tk)^3}{6} + \ddot{q}_{ik} \frac{(t-tk)^2}{2} + \dot{q}_{ik} \frac{(t-tk)}{1} + q_{ik}$$

-Vitesse généralisée: c'est une fonction parabolique du temps:

$$\dot{q}_i = a_{ik} \frac{(t-tk)^2}{2} + \ddot{q}_{ik} \frac{(t-tk)}{1} + \dot{q}_{ik}$$

-Accélération généralisée: c'est une fonction linéaire du temps:

$$\ddot{q}_i = a_{ik}(t-tk) + \ddot{q}_{ik}$$

avec:

$$a_{ik} = -\frac{12}{(\Delta t_k)^2} (\Delta q_{ik} - \dot{q}_{ik} \Delta t_k) + \frac{6}{(\Delta t_k)^2} \Delta \dot{q}_{ik}$$

$$\dot{q}_{ik} = \frac{6}{(\Delta t_k)^2} (\Delta q_{ik} - \dot{q}_{ik} \Delta t_k) - \frac{2}{\Delta t_k} \Delta \dot{q}_{ik}$$

où

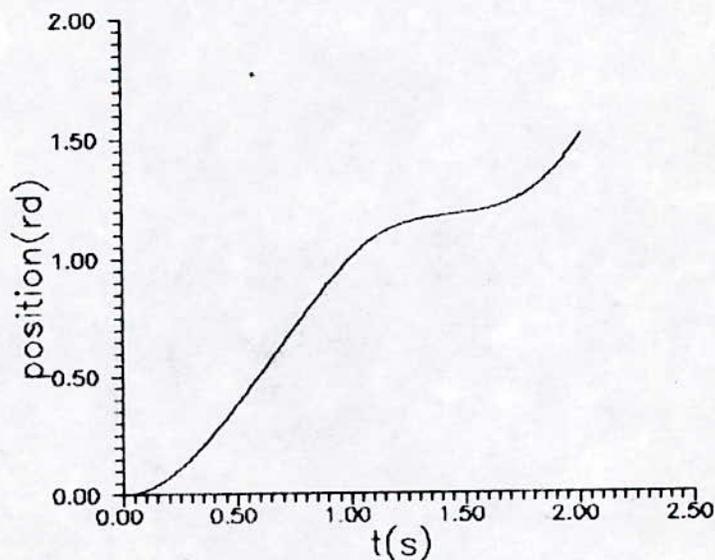
$$\Delta q_{ik} = q_{ik+1} - q_{ik} ; \Delta \dot{q}_{ik} = \dot{q}_{ik+1} - \dot{q}_{ik} ; \Delta t_k = t_{k+1} - t_k$$

et $q_i, \dot{q}_i, \ddot{q}_i$

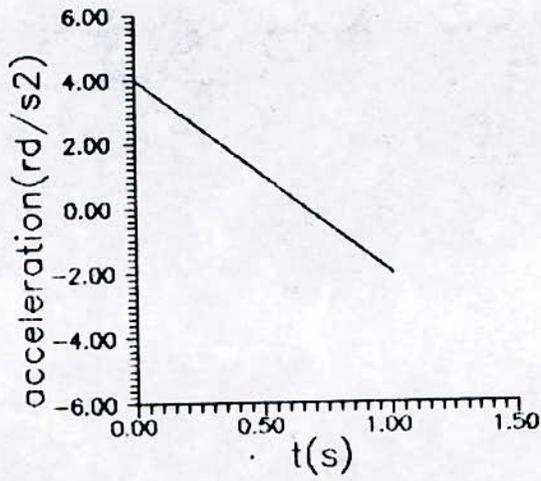
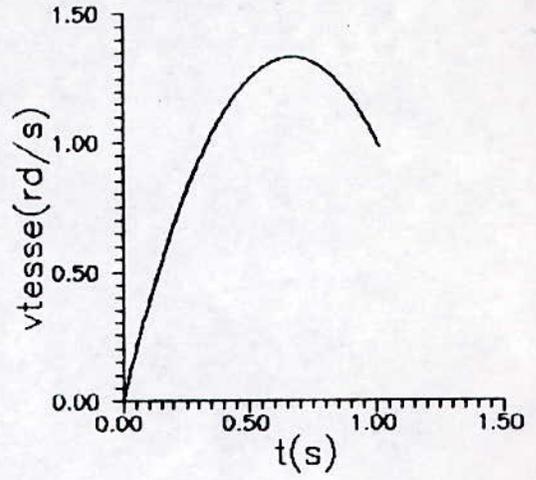
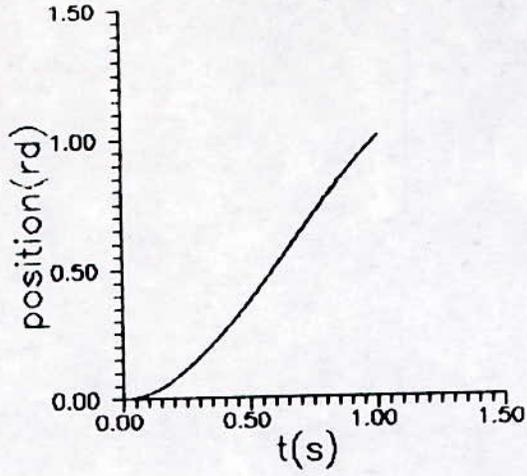
définissent les évolutions des coordonnées, vitesses et accélérations généralisées entre les instants t_k et t_{k+1}

Et voici des exemples de trajectoires:

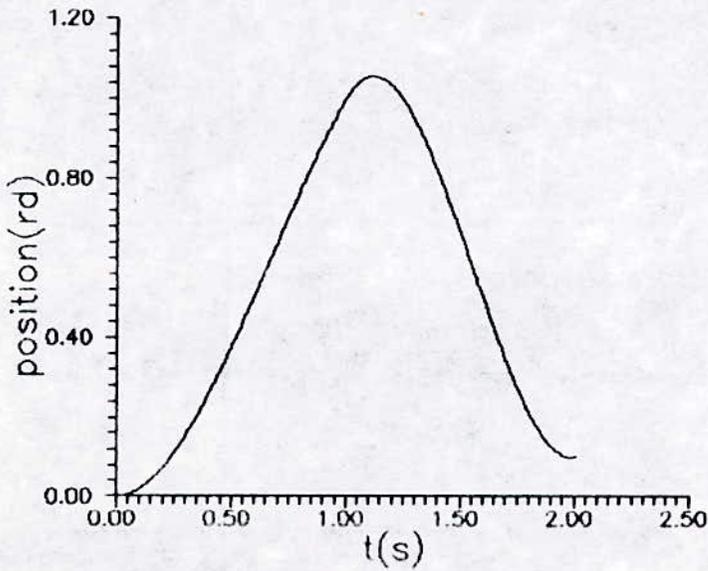
a.



b.



c.



III-7-Shéma global de simulation:

Le schéma global de simulation est développé dans la figure ci-dessous. Et en voilà une brève description:

-les modules du schéma de simulation correspondent à des unités en Turbo-Pascal:

.l'espace de travail étant le programme principal(LYAP.PAS ou POPOV.PAS), les données sont demandées et l'utilisateur les introduit.

.la generation de la trajectoire est assurée par l'unité GENERE.PAS, et les valeurs sont stockées dans le fichier genere.dat et ordonnées comme suit:temps, coordonnées, vitesses, accélération.

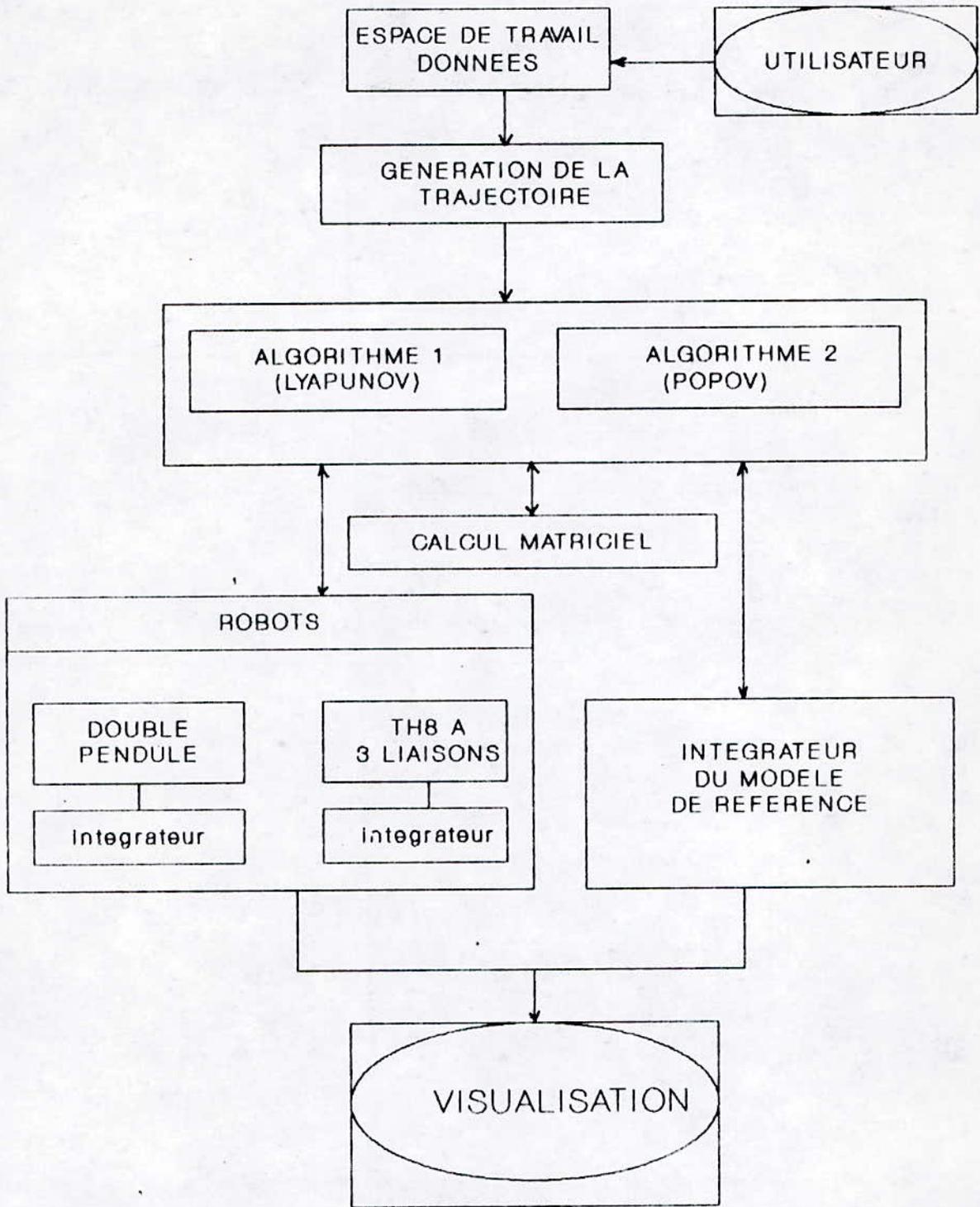
.l'unité INTEG.PAS assure l'intégration du modèle de référence.

.l'intégration des modèles de simulation est assurée par les unités DP.PAS(pour le R.M. double pendule) et TH8.PAS (pour le R.M.TH8 à trois liaisons).Chacune de ces unités contient le modèle de simulation du robot et son intégrateur.

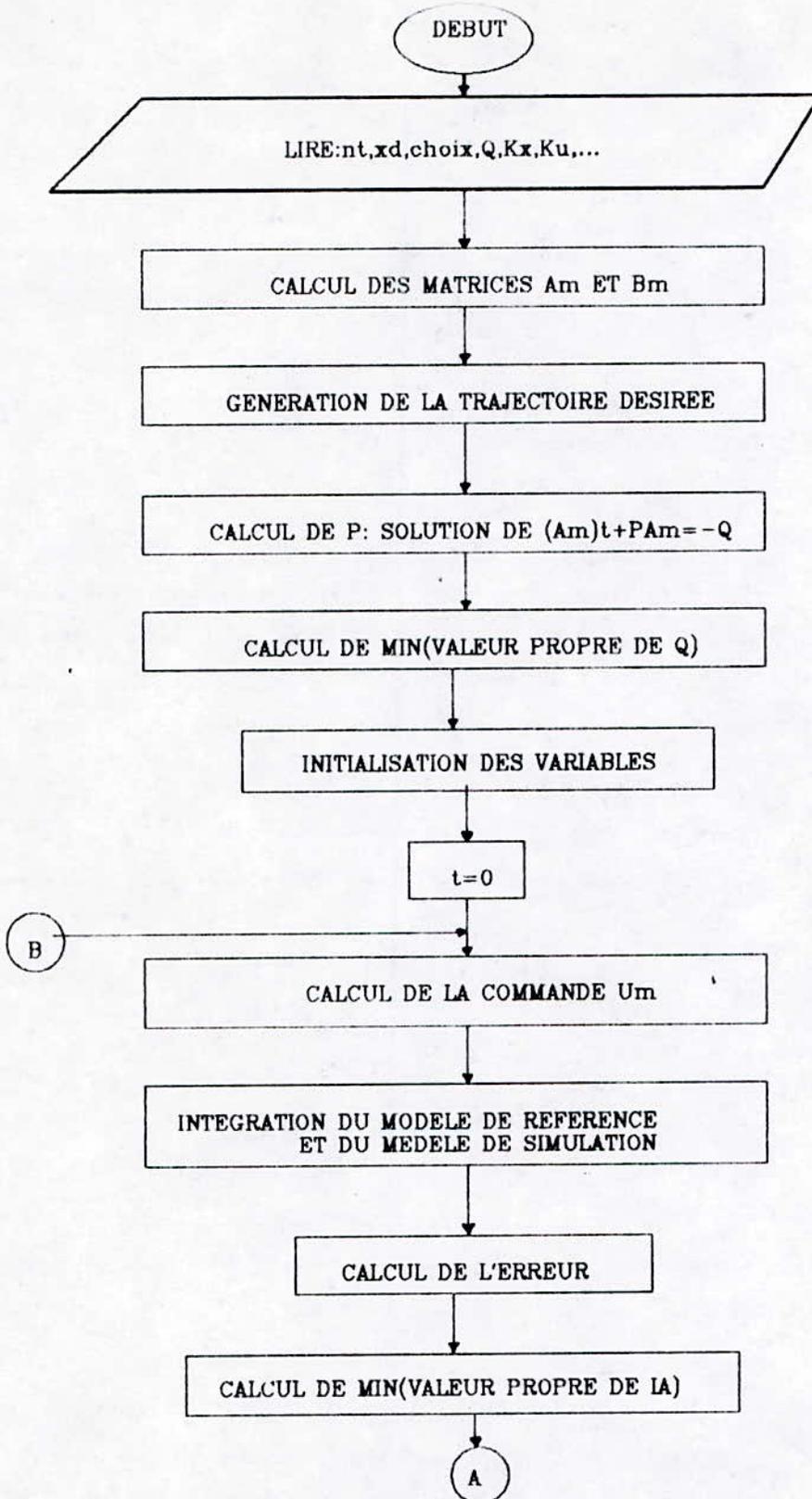
.l'unité VOIR.PAS assure la visualisation simultanée des positons désirées, du modèle de référence et du R.M.

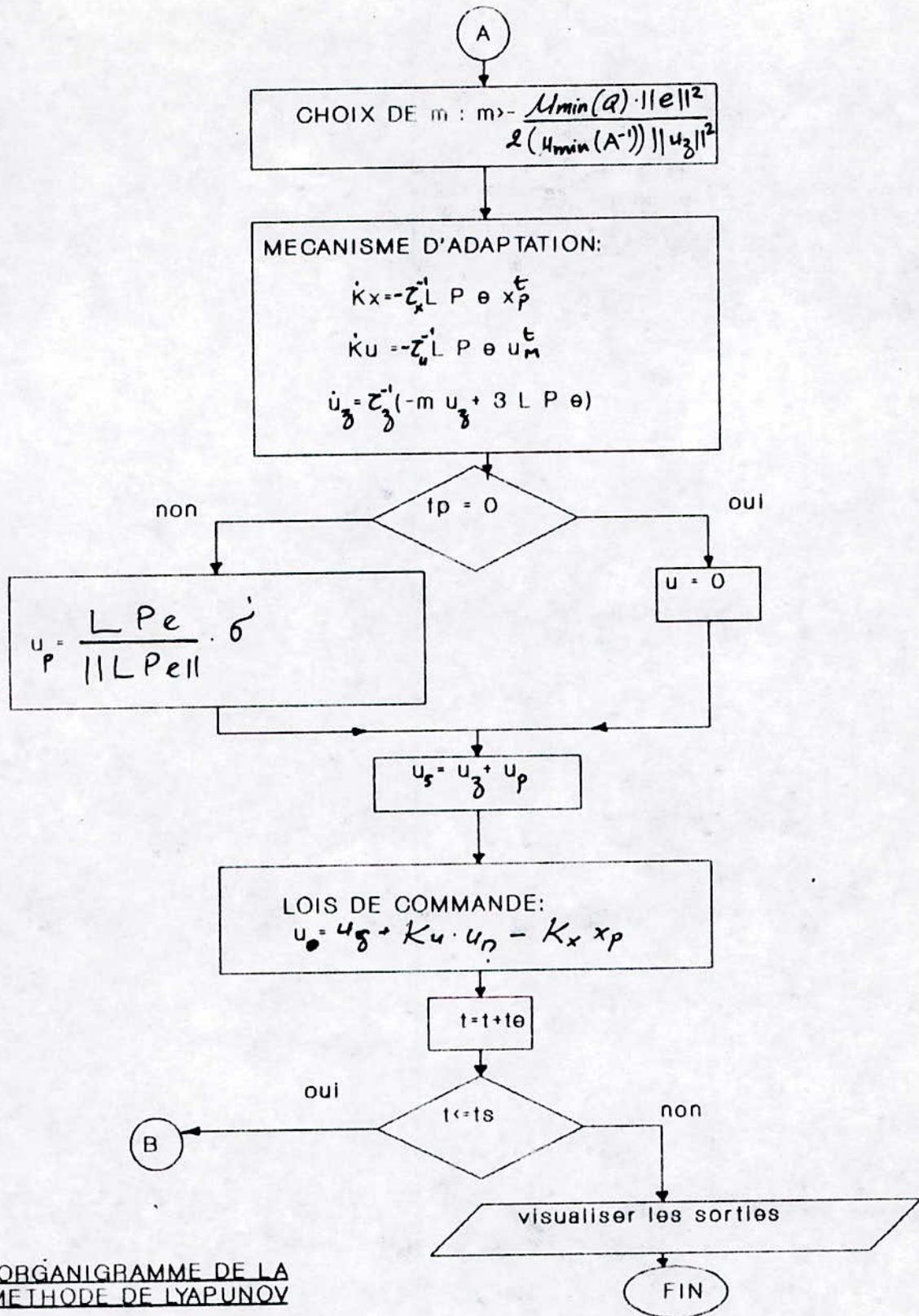
.une unité de calcul matriciel(MATOPER.PAS) est utilisée

par toutes les autres unités. Elle contient les opérations sur les matrices dont:la somme, le produit, l'inversion, la transposition, le calcul des valeurs propres, le calcul de la norme, produit par un scalaire, et autres.

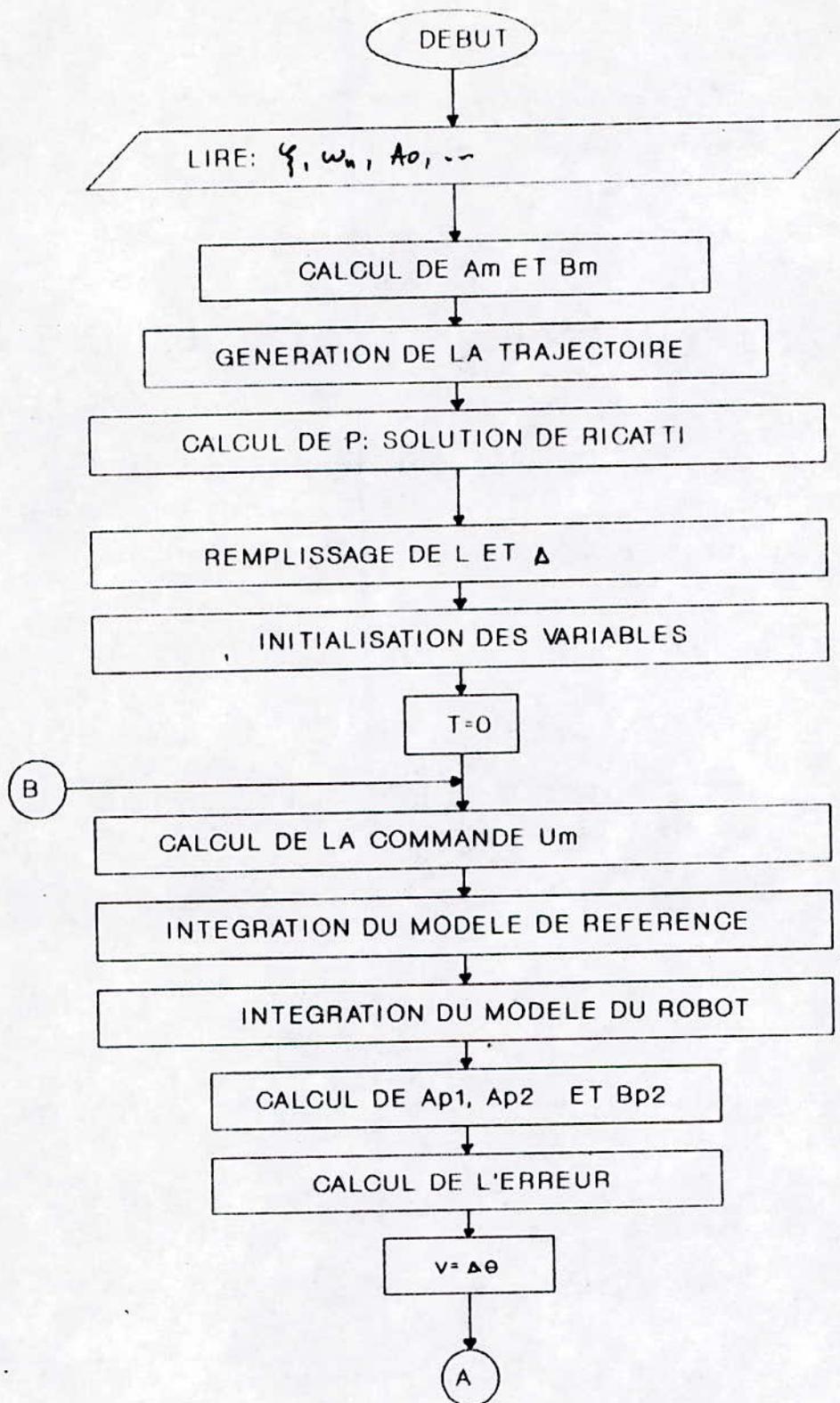


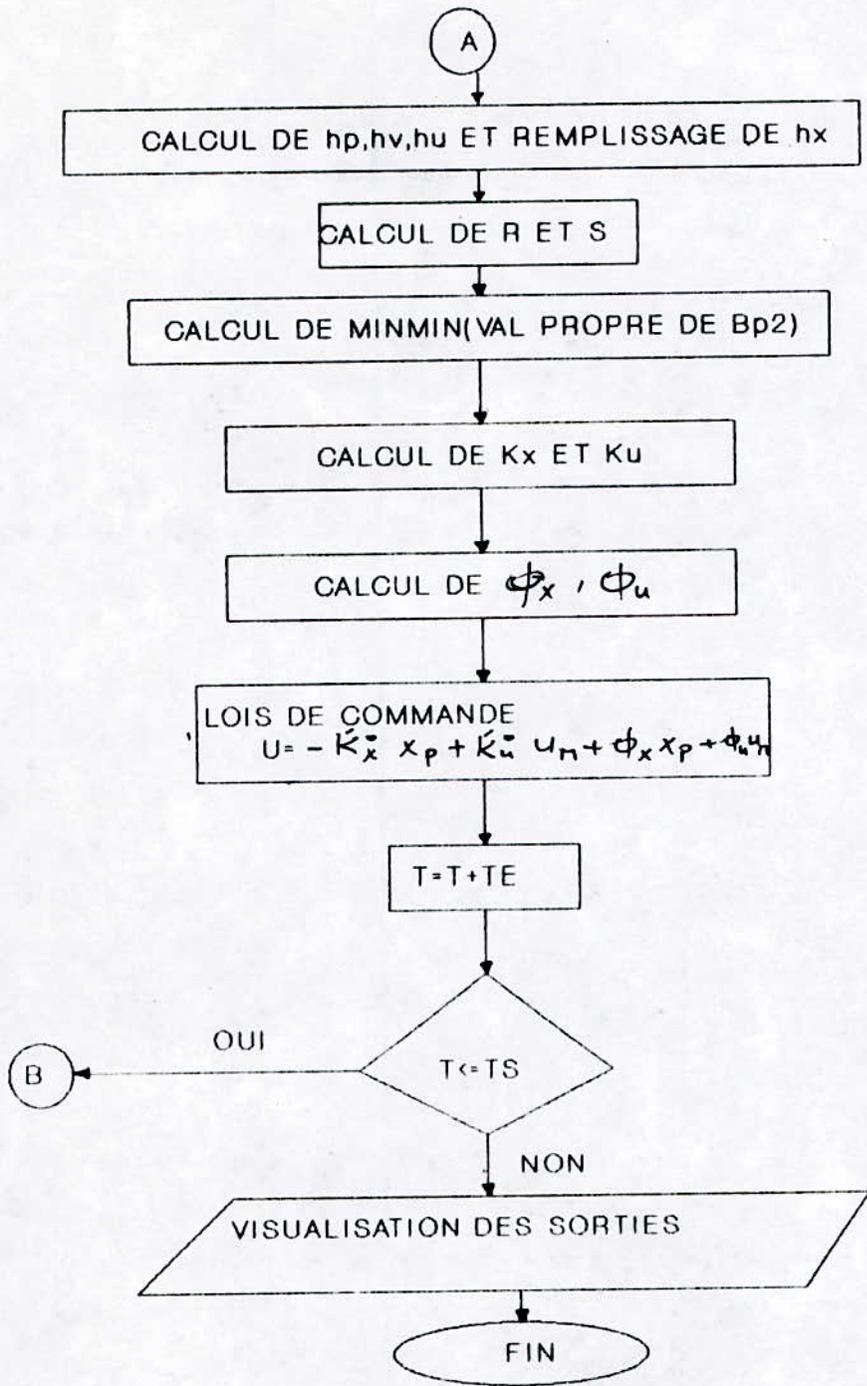
SCHEMA GLOBAL DE SIMULATION





ORGANIGRAMME DE LA METHODE DE LYAPUNOV





ORGANIGRAMME DE LA METHODE DE POPOV

Remarque:

Il est à noter que nous avons juxtaposé en amont de u_m , un bloc pour calculer celle-ci afin d'avoir la trajectoire du modèle de référence confondue à celle désirée (fig.III.5). Et comme ceci le R.M., en cherchant à suivre le modèle de référence, suivra la trajectoire désirée.

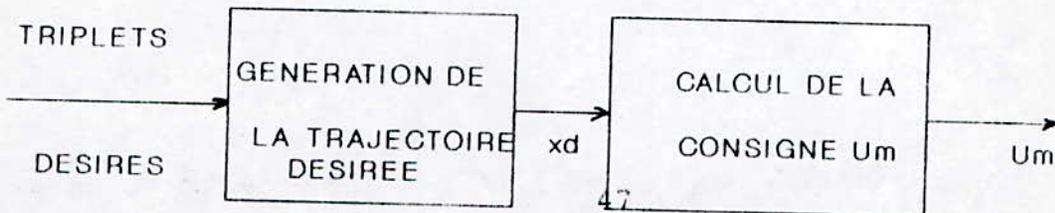


fig.III.5. Bloc de calcul de u_m .

Chapitre IV

CHAPITRE IV

PRESENTATION DES RESULTATS

VI-1- Suivi de trajectoire par la méthode de Lyapunov:

Le suivi de la trajectoire dépend de plusieurs paramètres dont les valeurs initiales de K_x, K_u, u_z .

Pour $\xi=0.7$, $w_n=1$

$K_x(1,1)=3, K_x(2,2)=K_x(1,3)=4, K_x(2,4)=5$

$K_u(1,1)=K_u(2,2)=1$

$u_z(1)=0.5, u_z(2)=0.8$

$Q(1,1)=Q(2,2)=2, Q(3,3)=Q(4,4)=1$

les autres éléments sont pris nuls.

On a obtenu les résultats donnés par les graphes ci-dessous.

On remarque que le R.M. suit le modèle de référence mais avec une erreur de poursuite. Mais l'essentiel est que le point initial et le point final sont atteints. Et on peut arriver à de meilleurs résultats si on procède à un réglage fin des différents paramètres.

Pour cela on va étudier l'influence de ces paramètres.

a-Influence de u_z initial:

En faisant varier la valeur initial du vecteur u_z , on a remarqué que lorsque celle-ci augmente, la position final n'est pas atteinte, elle est toujours inférieur à celle désirée. Parcontre si elle diminue, la position finale est supérieure à celle désirée.

b-Influence de la matrice Q:

La remarque faite sur cette matrice est que lorsque

elle augmente, la sortie du système s'éloigne de celle du modèle de référence en grandissant; et lorsqu'elle diminue le système ne peut pas poursuivre et par conséquent n'atteint pas la position désirée.

c-Influence de l'amortissement ξ :

Si elle est supérieure à 0.7 le processus poursuit mais avec un retard qui ne lui permet pas d'atteindre la position désirée. Et il va au-delà de celle-ci lorsqu'elle est inférieure à 0.7.

d-Influence de la pulsation propre ω_n :

Lorsque elle passe à une valeur supérieure à 1 le processus est tellement rapide qu'il se déstabilise et ne continue pas la poursuite.

e-Influence de K_x, K_u initiaux:

Pour ceux-ci il faut en trouver une bonne combinaison sinon le système ne suit pas la trajectoire désirée.

VI-2- Suivi de trajectoire par la méthode de Popov:

Dans cette méthode aussi le suivi de trajectoire dépend de quelques paramètres dont le plus important est la matrice nominale A_0 de la matrice d'inertie.

Pour les valeurs suivantes:

$$\xi=1, \omega_n = 2,$$

$$A_0(1,1)=0.25,$$

$$A_0(1,2)=A_0(2,1)=0.05,$$

$$A_0(2,2)=0.125,$$

$$Q(1,1)=Q(2,2)=10,$$

$$Q(3,3)=Q(4,4)=6,$$

les autres éléments de Q sont nuls.

On peut arriver à des résultats meilleurs en procédant à un réglage fin des paramètres. Pour ceci on a étudié l'influence de quelques paramètres.

a-Influence de Q:

Pour des valeurs moins importantes que celles données auparavant la sortie du R.M. suit celle du modèle de référence, mais avec une erreur plus importante. Par contre pour des valeurs élevées de Q le système s'éloigne beaucoup plus et la sortie est plus importante que celle désirée.

b-Influence de A0:

Ce paramètre paraît le plus important car dès qu'on le change légèrement la sortie du R.M. est nettement différente.

Remarque:

Pour les deux méthodes, les valeurs des paramètres dont on a étudié l'influence sont comparées à celles des paramètres pour lesquels on a présenté les résultats.

On a donc montré, pour les deux méthodes, l'influence des paramètres utilisés.

Il est à noter que dans l'erreur globale, il y a l'erreur d'intégration et l'erreur de poursuite.

On a obtenu les résultats suivants:

1. Méthode de Lyapunov:

— xd
 - - xm
 - - xp

figure IV. a

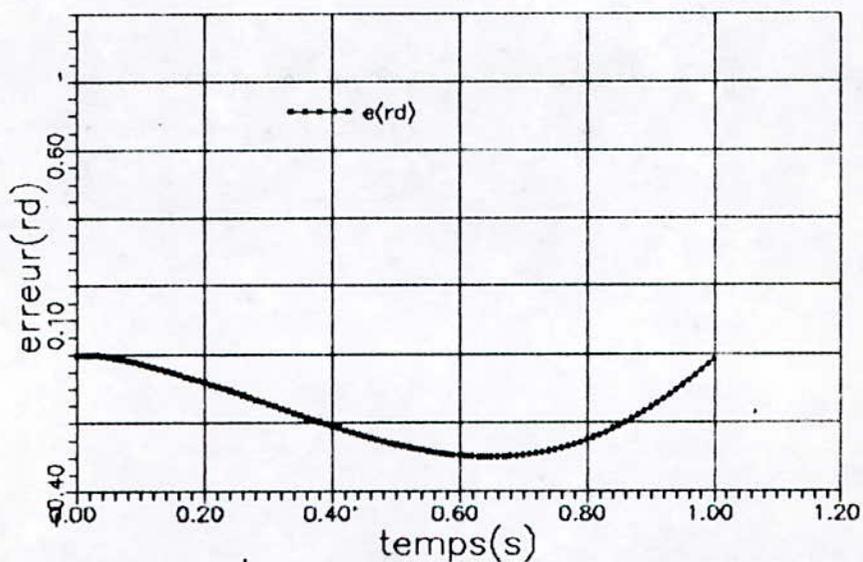
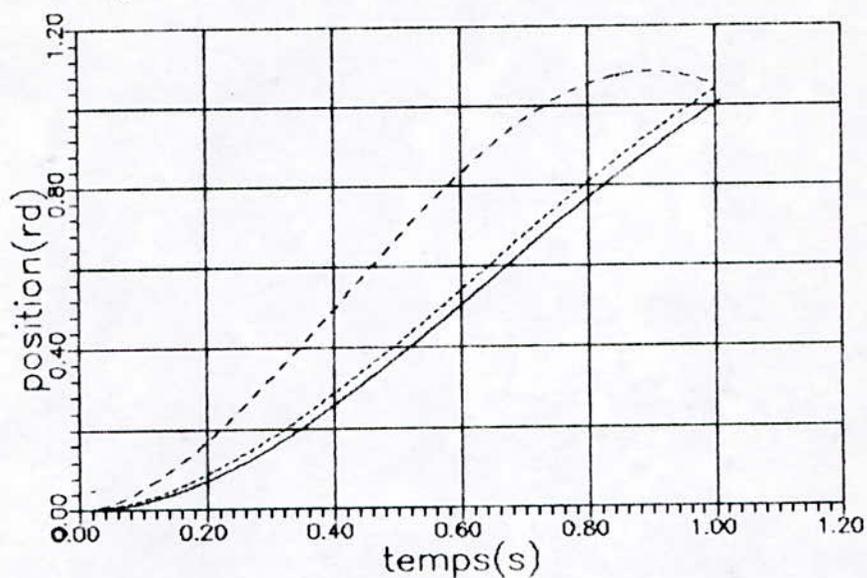


figure IV. b

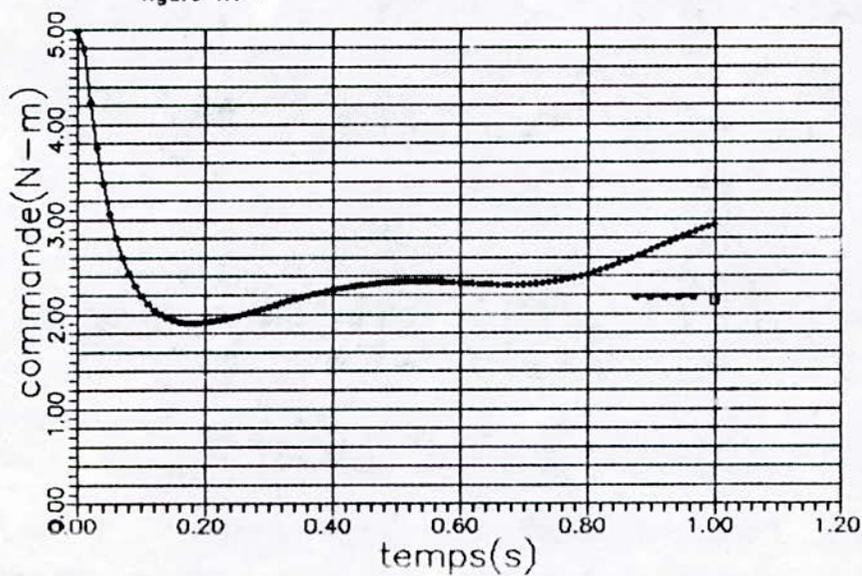
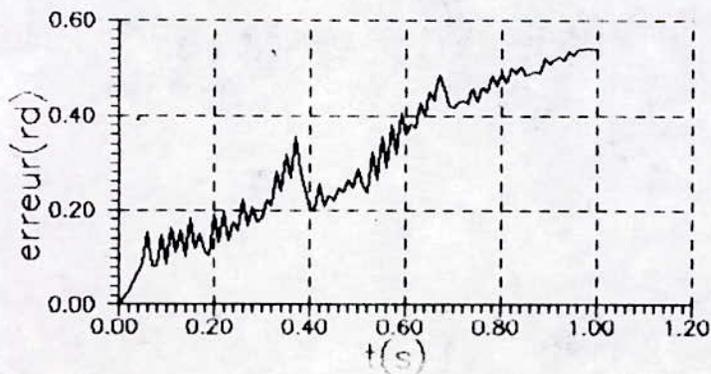
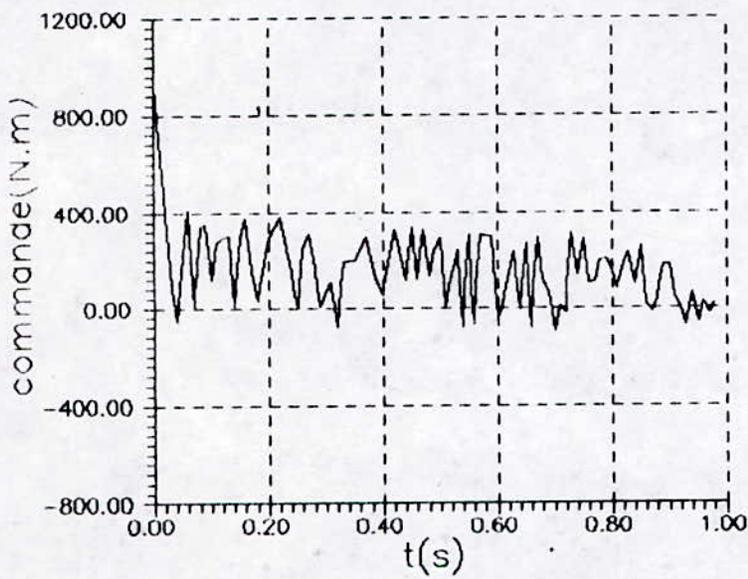
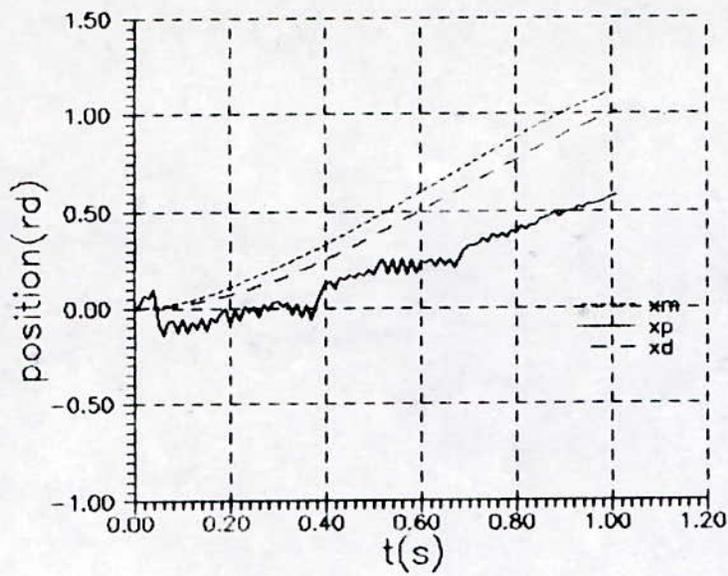


figure IV. c

2. Méthode de Popov:



Conclusion

CONCLUSION GENERALE

Le travail présenté dans ce mémoire a été consacré à la commande adaptative à modèle de référence appliquée aux robots manipulateurs.

Dans la première partie de ce travail (chapitre I), nous avons présenté une modélisation dynamique des robots manipulateurs, et la mise sous forme d'équation d'état. Dans la seconde partie (chapitre II), nous avons présenté la commande adaptative à modèle de référence appliquée aux robots manipulateurs, et ceci par la présentation de quelques approches de ce type de commande. Dans la troisième partie (chapitre III) nous avons donné les méthodes d'intégration utilisées, les modèles de simulation des robots utilisés, et les techniques de simulation. Enfin quelques résultats sont donnés dans la quatrième partie, illustrés d'interprétations et de remarques.

Les conclusions qui se dégagent de cette étude sont les suivantes:

- l'approche de commande adaptative MRAS reste une approche intéressante vu les considérations suivantes:

. elle ne nécessite pas l'identification du modèle dynamique,

. et son faible coût d'implantation.

- les difficultés rencontrées se situent au niveau de l'initialisation et du choix des valeurs des différents paramètres. D'où l'intérêt de l'étude en simulation car cela permet un choix judicieux des paramètres pour arriver

aux meilleurs performances possibles.

Les résultats obtenus montrent l'applicabilité de cette approche à la commande des robots manipulateurs.

Le logiciel que nous avons réalisé est ouvert à de nouvelles extensions par l'intégration d'autres algorithmes de commande et également d'autres robots manipulateurs de structures géométrique et mécanique variées.

Bibliographie

BIBLIOGRAPHIE:

- [1] Armici
-Turbo pascal version 4.
- [2] A.Balestino, G.De Maria et L.Sciavicco
-An adaptive model following control for robotic manipulators.
Journal of dynamic systems, measurement, and control. Sept 1983
- [3] B.Bouzouia
-Commande des robots manipulateurs: Identification des
paramètres et étude de stratégies adaptatives.
Rapport LAAS n°87.147
- [4] H.Bühler
-Conception des systèmes automatiques.
Presses Polytechniques Romandes
- [5] H.Bühler
- Réglages échantillonnés, Volume 2 "Traitement dans l'espace
d'état" Presses Polytechniques Romandes
- [6] S.Dubowsky et R.Kornhuber
-On the development of high performance adaptive control
algorithms for robotic manipulators.
Second international symposium of robotic research.
Japon août 1984
- [7] P.Fillatreau dirigé par B.Bouzouia
-Commande des robots manipulateurs
P.F.E. Toulouse 1987
- [8] Gorla b. et Renaud M.
-Modèles des robots manipulateurs-Application à leur commande
Cepadus 1984

[9]:Henry R.Asare et David G.Wilson

- Evaluation of three model reference adaptive control algorithms of robotic manipulators. Rockwell:International

[10]:K.Y.Lim et M.Eslami

- Robust adaptive controller designs for robot manipulator Systems. Journal of Robotics and automation, fevrier 1987

[11]:P.Conti

- Contribution à la commande dynamique-adaptative des robots manipulateurs.

Rapport LAAS n°87.350

[12]:B.Siciliano, B-S. Yuan et W.J.Book

- Model reference adaptive control of one link flexible arm. IEEE proceeding Decembre 1986.

[13]:Yoan D.Landau.

- Adaptive Control-Model reference approach

Annexes

*** ANNEXE A ***

unit genere;unit genere;unit genere;

IMPLEMENTATION

```

procedure generer(n:integer;ta:extended;var t:extended);
var t,tt:extended;
    qa,dda,a,q1,qt,dqt,ddqt,q11,qt2,qt3:array[1..6] of extended;
    tk:array[1..5] of extended;dq,dq,ddq:array[1..6,1..5] of extended;
    nt,k,i,k1:integer; des:text;
BEGIN
assign(des, c:\genera.dat);
rewrite(des);
(***** Lecture des donnees *****)
write('donner le nombre de triplets pour generer la trajectoire...');readln(nt);

for k:=1 to nt do
begin
writeln('tk',k); readln(tk[k]);
for i:=1 to n do
begin
writeln('q',k,i,'') et dq['',k,i,''];readln(q[k,i],dq[k,i]);
end;
end;
te:=tk[k];
(***** Boucle principale *****)
for k:=1 to nt-1 do
begin
k1:=k+1;

t:=tk[k];tt:=tk[k1]-tk[k];
for i:=1 to n do
begin
qa[i]:=q[k1,i]-q[k,i];dqa[i]:=dq[k1,i]-dq[k,i];
a[i]:=-12*(qa[i]-dq[k,i]*tt)/((sqr(tt)*tt)+6*dqa[i]/sqr(tt));
ddq[k,i]:=6*(qa[i]-dq[k,i]*tt)-2*dqa[i]/tt;
q1[i]:=q[k,i];

end;
write(des,t:9:5);
for i:=1 to n do write(des,q1[i]:9:5);
for i:=1 to n do write(des,dq[k,i]:9:5);
for i:=1 to n do write(des,ddq[k,i]:9:5);
writeln(des, '');
WHILE t<tk[k1] do
begin
t:=t+te;
write(des,t:9:5);
for i:=1 to n do
begin
qt1[i]:=a[i]*sqr(t-tk[k])*sqr(t-tk[k])/6;
qt2[i]:=ddq[k,i]*sqr(t-tk[k])/2;
qt3[i]:=dq[k,i]*(t-tk[k])+q1[i];
qt[i]:=qt1[i]+qt2[i]+qt3[i];(qt[i] est la position desiree de la liaison i)
dqt[i]:=a[i]*(sqr(t-tk[k]))/2+ddq[k,i]*(t-tk[k])+dq[k,i];(dqt[i] est la vitesse desiree de de la liaison i)
ddqt[i]:=a[i]*(t-tk[k])+ddq[k,i];
write(des,qt[i]:9:5);
end;
for i:=1 to n do write(des,dqt[i]:9:5);
for i:=1 to n do write(des,ddqt[i]:9:5);
writeln(des, '');
for i:=1 to n do
begin

```

```
begin
  {xd[i,1]:=qt[i];xd[i+n,1]:=dq[i];dxd[i,1]:=dqt[i];dxd[i+n,1]:=ddqt[i];}

  end;
end;
q[k,1]:=q[k1,1];dq[k,1]:=dq[k1,1];
(k:=k+i);
end;
close(des);
END;
BEGIN
END.
```

(*** ANNEXE B ***)

unit dp;

INTERFACE

uses matoper;

procedure rkdp(te,u1,u2:extended;var xp:tb;var a:tb);

IMPLEMENTATION

procedure rkdp(te,u1,u2:extended;var xp:tb;var a:tb);

const m1=0.5 ;m2=0.5;l1=0.5 ;l2=0.5 ;g=9.81;

var i,j:integer;

h11,h12,h13,h14,h21,h22,h23,h24,h31,h32,h33,h34,h41,h42,h43,h44,t:extended;

tie,a1,a2,b1,b2,a12,a13,a14,b12,b13,b14,a22,a23,a24,b22,b23,b24,(a1,a2,b1,b2),k0,k1,k2,s1,s2:extended;

function f3(a1,a2,b1,b2,g:extended):extended;

begin

k0:=m2*l1*(g*cos(a1-a2)*sin(a2)+l1*sin(a1-a2)*cos(a1-a2)*sqrt(b1)-l2*sin(a1-a2)*sqrt(b2));

k1:=u1-u2*l1*cos(a1-a2)/l2-(m1+m2)*g*l1*sin(a1);

k2:=((m1+m2)*sqrt(l1)-m2*sqrt(l1)*sqrt(cos(a1-a2)));

f3:=(k1+k0)/k2;

end;

function f4(a1,a2,b1,b2,g:extended):extended;

begin

(s1:=u2/m2*sqrt(l2)-cos(a1-a2)*(u1-(l1*(u2*cos(a1-a2)-sin(a1-b1)*sqrt(b1))+g*sin(a2))/l2);

s2:=(((m1+m2)*(l1)+m1*(l1)*sqrt(cos(a1-a2))/l2)-(l1*sin(a1-a2)*sqrt(b1)+g*sin(a2))/l2);

s1:=f3(a1,a2,b1,b2,g);

f4:=(u1-m2*l1*l2*sin(a1-a2)*sqrt(b2)-(m1+m2)*l1*(g*sin(a1)+l1*s1))/(m2*l2*l1*cos(a1-a2));

end;

function f1(a1,a2,b1,b2,g:extended):extended;

begin

f1:=b1;

end;

function f2(a1,a2,b1,b2,g:extended):extended;

begin

f2:=b2;

end;

REGIN

(a1:=1;a2:=1;b1:=1;b2:=1;)

a1:=xp[1,1];a2:=xp[2,1];b1:=xp[3,1];b2:=xp[4,1];

begin

h11:=f1(a1,a2,b1,b2,g);

h12:=f2(a1,a2,b1,b2,g);

h13:=f3(a1,a2,b1,b2,g);

h14:=f4(a1,a2,b1,b2,g);

a12:=a1+(te*h11)/2;

a22:=a2+(te*h12)/2;

b12:=b1+(te*h13)/2;

b22:=b2+(te*h14)/2;

h21:=f1(a12,a22,b12,b22,g);

h22:=f2(a12,a22,b12,b22,g);

h23:=f3(a12,a22,b12,b22,g);

h24:=f4(a12,a22,b12,b22,g);

a13:=a12+(te*h21)/2;

a23:=a22+(te*h22)/2;

b13:=b12+(te*h23)/2;

b23:=b22+(te*h24)/2;

h31:=f1(a13,a23,b13,b23,g);

h32:=f2(a13,a23,b13,b23,g);

h33:=f3(a13,a23,b13,b23,g);

h34:=f4(a13,a23,b13,b23,g);

a14:=a13+(te*h31)/2;

a24:=a23+(te*h32)/2;

b14:=b13+(te*h33)/2;

b24:=b23+(te*h34)/2;

```

h41:=f1(a14,a24,b14,b24,q);
h42:=f2(a14,a24,b14,b24,q);
h43:=f3(a14,a24,b14,b24,q);
h44:=f4(a14,a24,b14,b24,q);

a1:=a1+te*(h11+2*h21+2*h31+h41)/6;
a2:=a2+te*(h12+2*h22+2*h32+h42)/6;
b1:=b1+te*(h13+2*h23+2*h33+h43)/6;
b2:=b2+te*(h14+2*h24+2*h34+h44)/6;
(*****resultats*****)
a1:=a1-2*PI*trunc(a1/(2*pi));
a2:=a2-2*PI*trunc(a2/(2*pi));
b1:=b1-2*PI*trunc(b1/(2*pi));
b2:=b2-2*PI*trunc(b2/(2*pi));

(writeln(ti:8:4,a1:10:5,a2:10:5,b1:10:5,b2:10:5);)

end;
xp[1,1]:=a1;xp[2,1]:=a2;xp[3,1]:=b1;xp[4,1]:=b2;
(**** calcul de A ****)
a[1,1]:=(m1+m2)*sqr(l1);
a[1,2]:=m2*11*12*cos(a1-a2);
a[2,1]:=a[1,2];
a[2,2]:=m2*12*12;

( readln; )
END;

begin
end.

```

(** ANNEXE C **)

unit th831a;

INTERFACE:=0;

uses crt,MATOPER;

procedure calddq(x:tb;u1,u2,u3:extended;var ddq,d:tb);

PROCEDURE RKTH8(te,u1,u2,u3:extended;var x,d:TE);

(* ce module permet le calcul du mode'le de simulation du
robot manipulateur TH8 a' 3 liaisons *)

IMPLEMENTATION

procedure calddq(x:tb;u1,u2,u3:extended;var ddq,d:tb);

const

(* Acc'e'l'eration de la pesanteur *)

g=9.80;

var i:integer;

teta1,teta2,teta3,teta4,teta14,teta15,teta16,teta17,teta18,teta19,a2,r6:extended;

q:tb;(coordonne'e ge'ne'ralise'e)

dq:tb;(vitesse ge'ne'ralise'e)

{ddq: acc'e'l'eration de }

dq2:tb;(dq2[i]= dq[i]*dq[i])

dqq:tb;(dqq[i]=dq[i]*dq[i])

{d:tb;matrice d'inertie}

id:tb;(inverse de d)

h:tb;(vecteur des forces de coriolis et cetrifuges)

fp:tb;(vecteur des frottements secs et visqueux)

qr:tb;(vecteur des forces du'es a' la pesanteur)

u:tb;(tensions de commande)

r:tb;(rapports de reduction)

qa:tb;(gains des convertisseurs)

pk:tb;(const de couples)

oja:tb;(inerties propres des moteurs)

v1,w,v:tb;(vecteurs de travail)

(* l'expression calcule'e est :ddq=ia(v-h-fp-qr)*)

function signe(z:extended):extended;

var sz:extended;

begin

if z>0 then sz:=1

else sz:=-1;

signe:=sz;

end;

(* VALEURS NUMERIQUES DES PARAMETRES

Rapports de reduction

*)

begin

r(1,1):=8;

r(2,1):=20/0.06373;

r(3,1):= 20/0.0625;

r(4,1):=121.5;

r(5,1):=120;

r(6,1):=162;

(* Gains des convertisseurs *)

qa(1,1):=1.975;

qa(2,1):=0.896;

qa(3,1):=0.877;

qa(4,1):=0.65;

qa(5,1):=0.65;

```

h[2,1]:=q*(teta2+teta3);
h[3,1]:=-1*q[3,1]*(dq[1,1]*dq[1,1])*teta3+(dq[1,1]*dq[1,1])*teta4;

(* calcul du vecteur fp des forces due aux frottements secs et visqueux *)

fp[1,1]:=r[1,1]*r[1,1]*teta14*dq[1,1]+r[1,1]*teta15*signe(dq[1,1]);
fp[2,1]:=r[2,1]*r[2,1]*teta16*dq[2,1]+r[2,1]*teta17*signe(dq[2,1]);
fp[3,1]:=r[3,1]*r[3,1]*teta18*dq[3,1]+r[3,1]*teta19*signe(dq[3,1]);

(* calcul du vecteur v=u*ga*okc*r *)

v[1,1]:=u1*ga[1,1]*okc[1,1]*r[1,1];
v[2,1]:=u2*ga[2,1]*okc[2,1]*r[2,1];
v[3,1]:=u3*ga[3,1]*okc[3,1]*r[3,1];

(* calcul du vecteur acceleration ddq=ia*(v-h-fp-gr) *)

for i:= 1 to 3 do
    v1[i,1]:=v[i,1]-h[i,1]-fp[i,1];

    multmat(id,v1,ddq,3,3,1);
{ ecri_gat(ddq,3,1);}
{ READLN;}
END;

PROCEDURE RKTH8(te,u1,u2,u3:extended;var x,d:TB);

var
xr,z:tb;
h1,h2,h3,h4,k1,k2,k3,k4,l1,l2,l3,l4,hh1,hh2,hh3,hh4,kk1,kk2,kk3,kk4,l11,l12,l13,l14:extended;
(*** Declaration des fonctions *****)

function f4(x:tb):extended;

begin
calddq(x,u1,u2,u3,z,d);
f4:=z[1,1];
end;

function f5(x:tb):extended;
begin
(calddq(x,u1,u2,u3,z,d);)
f5:=z[2,1];
end;

function f6(x:tb):extended;
begin
(calddq(x,u1,u2,u3,z,d);)
f6:=z[3,1];
end;

function f1(x:tb):extended;
begin
f1:=x[4,1];
end;

function f2(x:tb):extended;
begin
f2:=x[5,1];
end;

```

```

function f3(x:tb):extended;
begin
  f3:=x[6,1];
end;

begin
  hh1:=f1(x);
  kk1:=f2(x);
  ll1:=f3(x);
  h1:=f4(x);
  k1:=f5(x);
  l1:=f6(x);
  xr[1,1]:=x[1,1]+hh1*te/2;
  xr[2,1]:=x[2,1]+kk1*te/2;
  xr[3,1]:=x[3,1]+ll1*te/2;
  xr[4,1]:=x[4,1]+h1*te/2;
  xr[5,1]:=x[5,1]+k1*te/2;
  xr[6,1]:=x[6,1]+l1*te/2;

  hh2:=f1(xr);
  kk2:=f2(xr);
  ll2:=f3(xr);
  h2:=f4(xr);
  k2:=f5(xr);
  l2:=f6(xr);
  xr[1,1]:=xr[1,1]+hh2*te/2;
  xr[2,1]:=xr[2,1]+kk2*te/2;
  xr[3,1]:=xr[3,1]+ll2*te/2;
  xr[4,1]:=xr[4,1]+h2*te/2;
  xr[5,1]:=xr[5,1]+k2*te/2;
  xr[6,1]:=xr[6,1]+l2*te/2;

  hh3:=f1(xr);
  kk3:=f2(xr);
  ll3:=f3(xr);
  h3:=f4(xr);
  k3:=f5(xr);
  l3:=f6(xr);
  xr[1,1]:=xr[1,1]+hh3*te;
  xr[2,1]:=xr[2,1]+kk3*te;
  xr[3,1]:=xr[3,1]+ll3*te;
  xr[4,1]:=xr[4,1]+h3*te;
  xr[5,1]:=xr[5,1]+k3*te;
  xr[6,1]:=xr[6,1]+l3*te;

  hh4:=f1(xr);
  kk4:=f2(xr);
  ll4:=f3(xr);
  h4:=f4(xr);
  k4:=f5(xr);
  l4:=f6(xr);
  x[1,1]:=x[1,1]+(te/6)*(hh1+2*hh2+2*hh3+hh4);
  x[2,1]:=x[2,1]+(te/6)*(kk1+2*kk2+2*kk3+kk4);
  x[3,1]:=x[3,1]+(te/6)*(ll1+2*ll2+2*ll3+ll4);
  x[4,1]:=x[4,1]+(te/6)*(h1+2*h2+2*h3+h4);
  x[5,1]:=x[5,1]+(te/6)*(k1+2*k2+2*k3+k4);
  x[6,1]:=x[6,1]+(te/6)*(l1+2*l2+2*l3+l4);

end;
begin
end;

```