

Table des matières

Résumé.....	III
Abstract.....	V
Table des matières.....	VII
Liste des Tableaux.....	IX
Liste des Figures.....	XI
Nomenclature.....	XV
Remerciements.....	XXI
Chapitre 1 Introduction.....	1
1.1 Situation actuelle.....	1
1.2 Problématique.....	1
1.3 Objectifs de la recherche.....	2
1.4 Organisation du mémoire.....	3
Chapitre 2 Modélisation des systèmes invariants dans le temps et l'application du modèle à paramètres variables.....	5
2.1 Introduction.....	5
2.2 Modélisation/identification d'une plate-forme mobile de soudage.....	7
2.3 Méthodes de réduction classiques.....	10
2.3.1 Skogestad « règle de la moitié ».....	10
2.3.2 Méthode de Padé.....	11
2.3.3 Méthode des fractions continues.....	12
2.3.4 Méthode d'approximation de Routh.....	14
2.3.5 Méthode de Schur.....	16
2.3.6 Méthode d'« Optimal Hankel norme approximation ».....	16
2.4 Exemple des réductions de l'ordre d'un modèle mathématique de la PFM subi d'une perturbation.....	17
2.5 Conclusion.....	20
Chapitre 3 Base de la commande par mode glissant à partir des équations d'état.....	21
3.1 Introduction.....	21
3.2 Équations d'état en forme de phase.....	21
3.3 Méthode des isoclines.....	23
3.4 Trajectoires de phase et leurs réponses temporelles.....	25
1) Trajectoires de phase en forme de col.....	26
2) Trajectoires de phase en forme de foyer.....	29
3) Trajectoires de phase en forme de nœud.....	31
3.5 Restitution des variables d'état avec un observateur.....	33

3.5.1	Introduction de l'observateur d'état général.....	33
3.5.2	Constriction de l'observateur d'état	36
3.6	Conclusion	42
Chapitre 4	Commande par mode glissant et les réponses temporelles dans le plan de phase	43
4.1	Introduction.....	43
4.2	Vérifier la stabilité du système et le point initial de stabilité	43
4.3	Commande par mode glissant d'un processus idéal.....	44
4.4	Commande par mode glissant d'un processus idéal au système étudié	47
4.5	Commande par mode glissant au système étudié.....	49
4.6	Conclusion	52
Chapitre 5	Plan d'état et les relations entre plans de phase et d'état.....	53
5.1	Introduction.....	53
5.2	Équations d'état en forme d'état	53
5.3	Transformation du plan de phase au plan d'état et calcul du plan d'état : pôles réels... 54	
5.3.1	Transformation géométrique du plan de phase au plan d'état.....	60
5.4	Transformation du plan de phase au plan d'état et calcul du plan d'état : cas pôles complexes conjugués.....	69
5.4.1	Transformation géométrique des trajectoires complexes conjuguées à la sphère de Riemann.....	75
5.5	Conclusion	77
Chapitre 6	Commande glissante dans le plan d'état.....	79
6.1	Introduction.....	79
6.2	Commande glissante et les réponses temporelles	79
6.3	Conclusion	86
Chapitre 7	Conclusions générales	87
	Références	89
	Annexe A : Plate-forme des calculs d'approximation des réponses temporelles.....	93
	Annexe B : Plate-forme des calculs des commandes par modes glissants	101
	Annexe C : Plate-forme des calculs des transformations géométriques.....	107
	Annexe D : Plate-forme des calculs des commandes glissantes	115

Liste des Tableaux

Tableau 2.3.1 Valeurs de α	15
Tableau 2.3.2 Valeurs de β	15
Tableau 3.3.1 Valeurs numériques de ρ	24
Tableau 3.3.2 Valeurs numériques de la pente d'isoclines.....	24

Liste des Figures

Figure 2.1.1	Ligne commutative et les quadrants du plan de phase	5
Figure 2.1.2	Exemple d'application de la commande par mode glissant idéal et réel, a) : Trajectoires du système dans le plan de phase, b) : Réponses temporelles du système..	6
Figure 2.2.1	Reproduction d'un système complexe : « Système commandé de soudage à l'arc avec robot mobile », a) : « Plate-forme mobile avec quelques capteurs », b) : « Système de propulsion de la PFM », c) : « Panneau Électrique de la PFM », d) : « Réseau sur micro-ordinateur comme une optimisation du régulateur »	7
Figure 2.2.2	Simulation de la réponse temporelle sans perturbations	8
Figure 2.2.3	Simulation des réponses temporelles avec des perturbations.....	9
Figure 2.4.1	Réponses temporelles de la PFM	18
Figure 2.4.2	Démonstration de l'application du modèle à paramètres variables.....	19
Figure 3.2.1	Schéma fonctionnel d'état	22
Figure 3.3.1	Trajectoires, en ligne brisée, sont trouvées	25
Figure 3.4.1	Familles de trajectoires de phase en forme de col (hyperbole)	26
Figure 3.4.2	Constructions les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ et les coordonnées rectangulaires $[v_1(t), v_2(t)]$ dans le plan de phase, a) : $\gamma \in \left(0 < \gamma < \frac{\pi}{2}\right)$, b) : $\gamma \in \left(\frac{\pi}{2} < \gamma < \pi\right)$, c) et d) : $\gamma = \frac{\pi}{2}$	28
Figure 3.4.3	Réponses temporelles du système avec les conditions initiales se placent sur les asymptotes stables.....	29
Figure 3.4.4	Familles de trajectoires de phase en forme de foyer (spirale).....	30
Figure 3.4.5	Réponses temporelles du système présenté en spirale	31
Figure 3.4.6	Familles de trajectoires de phase en forme de nœud.....	32
Figure 3.4.7	Réponses temporelles du système avec des conditions initiales d'asymptotes stables	33

Figure 3.5.1 Schéma de l'observation.....	36
Figure 3.5.2 Schémas de l'observateur d'état présenté en forme d'équation 3.22	37
Figure 3.5.3 Schémas de l'observateur d'état présenté en forme d'équation 3.23	37
Figure 3.5.4 Schéma d'observateur d'état dans le <i>Simulink</i>	38
Figure 3.5.5 Réponses temporelles mesurées et restituées par l'observateur	39
Figure 3.5.6 Erreur $e(t)$ en fonction du temps.....	39
Figure 3.5.7 Restitution de variable d'état $\hat{x}_1(t)$	40
Figure 3.5.8 Distance du temps retard.....	41
Figure 4.2.1 Schéma de l'observation.....	43
Figure 4.2.2 Vérification le point initial de stabilité	44
Figure 4.3.1 Trajectoire du système en forme de spirale (foyer)	45
Figure 4.3.2 Réponse temporelle du système sous forme de spirale (foyer)	45
Figure 4.3.3 Trajectoires des commandes par modes glissants idéales	46
Figure 4.3.4 Réponses temporelles des commandes par modes glissants d'un processus idéal.....	46
Figure 4.4.1 Démonstration de l'application de la commande par mode glissant d'un processus idéal au système étudié.....	47
Figure 4.4.2 Trajectoire de la commande construite par deux lignes commutatives	48
Figure 4.4.3 Réponse temporelle de la commande ci-dessus.....	48
Figure 4.5.1 Définition la ligne commutative	49
Figure 4.5.2 Trajectoire d'une commande par mode glissant	50
Figure 4.5.3 Réponse temporelle d'une commande par mode glissant.....	50
Figure 4.5.4 Limites de placement de la ligne commutative	51
Figure 4.5.5 Démonstration les trajectoires des commandes par modes glissants.....	51
Figure 4.5.6 Réponses temporelles des commandes proposées	52
Figure 5.2.1 Schéma fonctionnel d'état	53
Figure 5.3.1 Famille de trajectoires en forme de col dans deux plans, a) : plan de phase, b) : plan d'état.....	60
Figure 5.3.2 Élongation et rotation du point initial du système, a) : Élongation des coordonnées du point initial ($0 < \beta < \alpha < \frac{\pi}{2}$), b) : Rotation à sens horaire d'un angle	

α ($0 < \beta < \alpha < \frac{\pi}{2}$), c) : Élongation des coordonnées du point initial ($0 < \alpha < \beta < \frac{\pi}{2}$),	
d) : Rotation à sens horaire d'un angle α ($0 < \alpha < \beta < \frac{\pi}{2}$), e) : Élongation des	
coordonnées du point initial ($\alpha = \beta = \frac{\pi}{4}$), et f) : Rotation à sens horaire d'un angle	
α ($\alpha = \beta = \frac{\pi}{4}$)	61
Figure 5.3.3 Démonstration de la perspective et de l'angle de la rotation θ	63
Figure 5.3.4 Démonstration la bissectrice OC^{\wedge} et la bissectrice OC , a) : Bissectrice des	
asymptotes de phase OC^{\wedge} , b) : Bissectrice OC des coordonnées d'état $[v_1(t), v_2(t)]$	63
Figure 5.3.5 Définition de l'angle θ par la relation géométrique	64
Figure 5.3.6 Définition de l'angle θ par la relation géométrique lorsque $\gamma \in \left(\frac{\pi}{2} < \gamma < \pi\right)$.	66
Figure 5.3.7 Démonstration de la projection du point initial dans le plan de phase, a) :	
Projection du point initial d'état $(v_1(0), v_2(0))$ dans le plan de phase, b) : Point initial	
$(v_1(0), v_2(0))$ dans le plan d'état.....	67
Figure 5.3.8 Démonstration de l'élongation des coordonnées du point initial de phase	67
Figure 5.3.9 Démonstration du plan d'état et du plan de phase dans l'espace avec pôles	
réels : $s_1 = 2$ et $s_2 = -1$	68
Figure 5.3.10 Démonstration du plan d'état et du plan de phase dans l'espace avec pôles	
réels : $s_1 = 0,5$ et $s_2 = -0,5$	69
Figure 5.4.1 Une portion de l'arc de spirale du système, où : $R(t)$ est la distance à l'origine,	
et $\Theta(t)$ est l'angle polaire.....	72
Figure 5.4.2 Famille de trajectoires en forme de foyer (spirales), a) : Trajectoires dans le	
plan de phase, b) : Trajectoires réelles dans le plan $[z_1(t), z_2(t)]$, c) : Trajectoires	
complexes conjuguées $(\tilde{v}_1(t), \tilde{v}_2(t))$ présentées par la partie réelle	
$\text{Re}[f(R(t), \Theta(t))]$ et la partie imaginaire $\text{Im}[f(R(t), \Theta(t))]$	74
Figure 5.4.3 Définition la sphère de <i>Riemann</i>	75

Figure 5.4.4 Démonstration la transformation géométrique à la sphère de <i>Riemann</i>	77
Figure 6.2.1 Démonstration des convergences de trajectoires, a) : Convergence des trajectoires hyperboles dans le plan d'état, b) : Projection la sphère de <i>Riemann</i> sur la surface $[v_{R1}, v_{R2}]$, c) : Convergence des trajectoires spirales sur la sphère de <i>Riemann</i> , d) : Construction de l'hypersphère commutative.....	81
Figure 6.2.2 Trajectoire en zigzag d'une commande glissante, où : la surface jaune est le plan d'état, la sphère bleue est la sphère de <i>Riemann</i>	81
Figure 6.2.3 Agrandissement de cette trajectoire.....	82
Figure 6.2.4 Réponse temporelle obtenue par l'application de la commande glissante	83
Figure 6.2.5 Limites de placement de la ligne commutative dans le plan d'état, a) : Période d'échantillonnage $t_e = 0,05s$, b) : Période d'échantillonnage $t_e = 0,10s$	84
Figure 6.2.6 Réponses temporelles du système, a) : Réponses temporelles trouvées avec l'application des commandes par modes glissants, b) : Réponses temporelles trouvées avec l'application des commandes glissantes	85

Nomenclature

A :	Matrice d'évolution du système du plan de phase
A_p :	Matrice d'évolution du système du plan d'état
$A_k(s)$:	Dénominateur de la fonction de transfert d'ordre réduit (Utilisation la méthode de Routh)
$\begin{bmatrix} A_{11} \dots A_{1n} \\ \vdots \\ A_{n1} \dots A_{nn} \end{bmatrix}$:	Matrice de Routh
$a_0 \dots a_m \dots a_n$:	Paramètres du numérateur de la fonction de transfert
$a_0^0 \dots a_n^n$:	Paramètres des calculs dans le tableau α
$\begin{cases} ang_x \\ ang_y \\ ang_z \end{cases}$:	Angles dans les matrices de rotations
B :	Matrice d'application de la commande du plan de phase
B_p :	Matrice d'application de la commande du plan d'état
$B_j(s)$:	Numérateur de la fonction de transfert d'ordre réduit (Utilisation la méthode de Routh)
$B_{01} \dots B_{2k,k+1}$:	Éléments dans la matrice de la transformation (Utilisation la méthode des fractions continues)
$b_0 \dots b_n$:	Paramètres du dénominateur de la fonction de transfert
$b_0^0 \dots b_n^n$:	Paramètres des calculs dans le tableau β
C :	Matrice d'observation du plan de phase
C_p :	Matrice d'observation du plan d'état
c :	Constante
$c_0 \dots c_n$:	Paramètres dans l'équation de la réduction (Utilisation la méthode de Padé)

$d_0 \dots d_j :$	Paramètres du numérateur de la fonction de transfert d'ordre réduit (Utilisation la méthode de Padé et la méthode des fractions continues)
$E(s) :$	Entrée du système/modèle dans la fonction de transfert
$e(t) :$	Erreur entre l'observateur d'état et le système/modèle
$\dot{e}(t) :$	Dérivée d'erreur
$e_0 \dots e_k :$	Paramètres du dénominateur de la fonction de transfert d'ordre réduit (Utilisation la méthode de Padé et la méthode des fractions continues)
$F_1(s) \dots F_n(s) :$	Équations des fractions continues
$G(s) :$	Fonction de transfert du système/modèle
$G_0(s) :$	Fonction de transfert d'ordre réduit à premier ordre
$G_C(s) :$	Fonction de transfert du correcteur
$G_{SL}(s) :$	Fonction de transfert du modèle à paramètres variables
$h_0, h_1 \dots h_n :$	Paramètres du numérateur de la fonction de transfert en parallèle
$K :$	Paramètre du numérateur de la fonction de transfert à premier ordre
$K_e :$	Élongation des coordonnées du point initial
$K_p, K_i :$	Paramètres du correcteur
$k :$	Pente de la ligne commutative
$\mathbf{L} :$	Gain \mathbf{L}
$LC :$	Ligne commutative
$L_d :$	Distance entre le point initial de stabilité et le point de la restitution
$l_1, l_2 :$	Éléments dans le gain \mathbf{L}
$\mathbf{M} :$	Matrice modale
$\mathbf{M}^{-1} :$	Matrice inversée de la matrice modale
$N :$	Matrice dans la formule de Bass-Gura
$P_{Reel}(t) :$	Partie réelle des trajectoires complexes conjuguées
$P_{Imag}(t) :$	Partie imaginaire des trajectoires complexes conjuguées
$p_{11}, p_{12} \dots p_{34} :$	Éléments dans les vecteurs propres

$R_k(s)$:	Fonction de transfert d'ordre réduit
$R_r(\alpha)$:	Rotation à sens horaire d'un angle α
R :	Rayon de la sphère de Riemann
R_c :	Rayon du cercle estimé à partir du point initial de stabilité
$R(t)$:	Rayon polaire
$\dot{R}(t)$:	Dérivée du rayon polaire
$R(0)$:	Rayon polaire initial
$\begin{cases} R_x(ang_x) \\ R_y(ang_y) \\ R_z(ang_z) \end{cases}$:	Matrices de rotation à sens inverse d'horaire
$r_0 \dots r_k$:	Paramètres dans l'équation de la réduction (Utilisation la méthode de Padé)
$s_1, s_2 \dots s_n$:	Pôles du système
T :	Paramètre du dénominateur de la fonction de transfert à premier ordre
$T_p^{-1}(\theta)$:	Transformation inverse de la perspective avec l'angle de la rotation θ
t_e :	Période d'échantillonnage
t_m :	Temps de la réponse temporelle mesuré
t_r :	Temps de la réponse temporelle du système/modèle
$\mathbf{u}(t)$:	Entrée du système/modèle (Multi)
$u(t)$:	Entrée du système/modèle (Single)
$v_1(t) \dots v_n(t)$:	Variables d'états du plan d'état
$\dot{v}_1(t) \dots \dot{v}_n(t)$:	Dérivées des variables d'état du plan d'état
$v_1(0), v_2(0)$:	Conditions initiales dans le plan d'état
$v_{R1}(t) \dots v_{R3}(t)$:	Coordonnées de la sphère de Riemann
$\hat{v}_1(t), \hat{v}_2(t)$:	Coordonnées obliques dans le plan de phase
$\tilde{v}_1(t), \tilde{v}_2(t)$:	Variables d'états du plan d'état avec pôles complexes conjugués

$\dot{\tilde{v}}_1(t), \dot{\tilde{v}}_2(t) :$	Dérivées des variables d'état du plan avec pôles complexes conjugués
$\tilde{v}_1(0), \tilde{v}_2(0) :$	Conditions initiales des trajectoires complexes conjuguées
$W :$	Matrice dans la formule de Bass-Gura
$\mathbf{x}(t) :$	Variables d'état du système/modèle
$\hat{\mathbf{x}}(t) :$	Variables d'état de l'observateur d'état
$\dot{\mathbf{x}}(t) :$	Dérivées des variables d'état du système/modèle
$\dot{\hat{\mathbf{x}}}(t) :$	Dérivées des variables d'état de l'observateur d'état
$x_1(t) \cdots x_n(t) :$	Variables d'état du plan de phase
$\dot{x}_1(t) \cdots \dot{x}_n(t) :$	Dérivées des variables d'état du plan de phase
$x_1(0) \cdots x_n(0) :$	Conditions initiales dans le plan de phase
$x_1(0), x_2(0), x_z(0) :$	Point initial d'état dans l'espace
$\hat{x}_1(t), \hat{x}_2(t) :$	Variables d'état de l'observateur d'état de second ordre
$\dot{\hat{x}}_1(t), \dot{\hat{x}}_2(t) :$	Dérivées des variables de l'observateur d'état de second ordre
$Y(s) :$	Sortie du système/modèle dans la fonction de transfert
$y(t) :$	Sortie du système/modèle
$\hat{y}(t) :$	Sortie de l'observateur d'état
$z_1(t), z_2(t) :$	Variables d'états de la surface aux valeurs réelles
$\dot{z}_1(t), \dot{z}_2(t) :$	Dérivées des variables d'état de la surface aux valeurs réelles
$z_1(0), z_2(0) :$	Conditions initiales dans la surface aux valeurs réelles
$\alpha :$	Angle entre l'asymptote de phase et la coordonnée $x_j(t)$
	Partie réelle des pôles complexes conjugués
$\alpha_1 \dots \alpha_n :$	Éléments dans le tableau de α
$\beta :$	Angle entre l'asymptote de phase et la coordonnée $x_j(t)$
	Partie imaginaire des pôles complexes conjugués
$\beta_1 \dots \beta_n :$	Éléments dans le tableau de β

δ :	Partie imaginaire des paramètres h_0, h_1
Φ :	Angle de la perspective
φ :	Angle dans ΔP_1NM
γ :	Angle entre les asymptotes de phase / Partie réelle des paramètres h_0, h_1
$\lambda_1, \lambda_2 \dots \lambda_n$:	Pôles d'observateur d'état
λ_1, λ_2 :	Valeurs propres de la matrice A
$\vec{\lambda}_1, \vec{\lambda}_2$:	Vecteurs propres de la matrice A
$\Theta(t)$:	Angle polaire
$\dot{\Theta}(t)$:	Dérivée d'angle polaire
$\dot{\Theta}(0)$:	Angle polaire initial
θ :	Angle de la rotation
θ_0 :	Paramètre du retard (Utilisation la méthode Skogestad « règle de la moitié »)
ρ :	Dérivée : $\frac{dx_2(t)}{dx_1(t)}$
τ :	Temps retard
τ_1, τ_2 :	Paramètres du dénominateur de la fonction de transfert d'ordre réduit (Utilisation la méthode Skogestad « règle de la moitié »)
ω :	Angle dans la figure 5.12

Remerciements

Tout d'abord, je tiens à remercier mon superviseur le professeur Stanislaw Tarasiewicz pour m'avoir donné la chance de réaliser ce projet de maîtrise. J'aimerais le remercier pour toute l'aide, les cours ainsi que les disponibilités qu'il m'a offertes généreusement durant toute la maîtrise.

Je voudrais remercier Docteur Feng Ding qu'il m'a énormément guidé et aidé à travers tout mon apprentissage du domaine. De même, je remercie aussi mon collègue N'zi Yoboué Guillaume, mon ami monsieur Qiwei He et sa femme Catherine Cliche. Ils m'ont aussi aidé à travers l'ensemble de cette étude. Merci aussi au professeur Benoît Lévesque qui m'a aidé pendant le cours d'Hydraulique et Pneumatique Industrielle.

Je profite aussi de l'occasion pour remercier ma famille au complet. Ils ont toujours été là pour m'encourager malgré la distance qui nous sépare. De plus, je souhaite aussi remercier mon oncle Hu Zhang, ma tante Xiaoyuan Chen et ma cousine Ke Yuan Zhang qui ont été là pour soutenir ma vie au Québec tant mon étude que ma vie personnelle.

Je voudrais également remercier le département de génie mécanique et les autres départements qui m'ont aidé pendant mon étude à l'université Laval.

En fin, je tiens à remercier encore une fois mon superviseur le professeur Tarasiewicz, pour toutes ses connaissances et ses expériences qui ont su satisfaire ma curiosité intellectuelle et qui m'inspire à avance continuellement, peu importe les circonstances de la vie.

Chapitre 1 Introduction

1.1 Situation actuelle

Dans l'application industrielle, la commande par PID [1, 2, 3] est une méthode souvent employée pour les asservissements. L'histoire des régulateurs PID est très longue. Les premiers régulateurs de type centrifuge sont apparus vers les années 1750 et sont utilisés pour régler la vitesse des moulins à vent. Après que les régulateurs sont créés, ils sont utilisés de nombreuse commande de procédés industriels, en moteurs, dans le monde de l'automobile, en robotique, etc. Il y a plusieurs raisons d'utiliser cette commande. Premièrement, le régulateur PID est très simple à mettre en place et s'avère efficace pour des systèmes réels. Même pour un système qui ne nécessitant pas de très grandes précisions dans l'asservissement, il est possible d'obtenir rapidement des valeurs des paramètres du régulateur PID. Deuxièmement, il y a des méthodes mathématiques avancées qui fournissent des techniques pour obtenir des valeurs des paramètres idéaux pour effectuer la commande des systèmes en particulier. Enfin, la commande par PID peut s'avérer inefficace pour les systèmes qui contiennent des perturbations ou les systèmes qui ne sont pas linéaires. Par conséquent, l'utilisation de la commande par PID est la méthode rapide et efficace.

1.2 Problématique

Plusieurs raisons d'application de la commande par PID sont présentées ci-dessus, mais cette commande aussi a un certain nombre d'inconvénients. Tout d'abord, les paramètres du régulateur sont immuables après les définitions. Dans ce cas, lorsqu'on applique cette commande, le système utilise plus de temps pour atteindre la valeur de la consigne. Par ailleurs, lorsqu'il arrive des perturbations, il est difficile d'assurer la stabilité du système. Enfin, il faut avoir beaucoup d'expérimentations pour obtenir rapidement des valeurs des paramètres de régulateur correctes dans un système en particulier. Par conséquent, on propose de faire l'application de la commande par mode glissant dans le système parce qu'il s'agit d'une commande optimale pour assurer la stabilité du système, en même temps de minimiser le temps de la réponse temporelle. L'idée de cette commande a été proposée en 1960 par *Emel'yanov* et *Barbashin*. Après les publications du livre par *Itkis* en 1976 [4] et un article par *Utkin* en 1977 [5], cette commande est devenue reconnue au monde. Remarquons aussi à ce sujet que grâce à l'article « *State Space Multiple Leaves for Linear Time-Varying Systems*

attempting Sliding Mode Controllers : A original approach » [12], cette commande est redéfinie et allongée dans le plan d'état par les familles multiples d'état.

1.3 Objectifs de la recherche

Dans ce mémoire, on a étudié une possibilité d'appliquer une nouvelle commande comme la commande glissante dans le plan d'état où le plan de phase et la commande par mode glissant sont la base de la transformation. Une plate-forme mobile de soudage (PFM) est choisie comme le système étudié pour appliquer la commande par mode glissant et cette nouvelle commande, la commande glissante. Dans ce cas, les objectifs de ce projet sont ceux-ci :

1. La modélisation du système étudié avec un modèle pour appliquer la commande par mode glissant et la commande glissante.
2. La recherche sur la façon de transformer le plan de phase au plan d'état pour construire la commande glissante.

Pour ce fait, d'abord, un modèle à paramètres variables doit être construit pour modéliser correctement notre système étudié. La structure de ce modèle contient un correcteur PI, une fonction de transfert réduite à premier ordre et une boucle retour. Grâce aux paramètres du correcteur ajustables, on a la possibilité d'appliquer la commande par mode glissant et la commande glissante. Ensuite, une plate-forme informatique est construite pour faire la transformation et voir l'effet de l'application de ces commandes. La plate-forme informatique est construite en trois niveaux :

1. La démonstration de l'application de la commande par mode glissant dans le plan de phase.
2. La démonstration des transformations géométriques du plan de phase au plan d'état pour trouver la base de la commande glissante.
3. La démonstration de l'application de la commande glissante dans le plan d'état.

Finalement, avec l'aide de cette plate-forme informatique, on a obtenu la nouvelle commande en forme explicite, la commande glissante qui est construite dans le plan d'état.

1.4 Organisation du mémoire

Le chapitre 2 introduit l'idée de la modélisation/identification pour construire un modèle du système. La fonction de transfert du système dont on a besoin. Elle est d'ailleurs de second ordre pour appliquer de la commande par mode glissant et la commande glissante. La méthode de la réduction de l'ordre des fonctions de transfert est utilisée lorsqu'on fait l'identification de notre système étudié. Dans ce cas, l'introduction de ces méthodes de la réduction est présentée dans ce chapitre. Cependant, on a trouvé que les réductions de fonction de transfert n'approximent pas correctement les réponses temporelles mesurées. Par conséquent, la construction d'un modèle à paramètres variables est proposée pour résoudre ce problème.

Le chapitre 3 introduit le plan de phase et la base de la commande par mode glissant. Cette base consiste en deux parties : premièrement, les familles de trajectoires de phase et deuxièmement, la restitution du point initial de stabilité. On a trouvé que les variables d'état dans le plan de phase sont inséparables, les trajectoires de phase sont en forme numérique. Alors, la méthode d'isoclines est introduite dans ce chapitre pour trouver les trajectoires de phase. On fait aussi l'introduction de l'observateur d'état pour faire la restitution du point initial de stabilité.

Le chapitre 4 présente la construction et l'application de la commande par mode glissant dans le plan de phase. La démonstration de cette commande est présentée dans plusieurs possibilités. Dans ce cas, on a trouvé qu'il existe un temps retard lorsque cette commande est appliquée dans notre système étudié. De nombreux chercheurs tentent de résoudre ce problème. *Wu-Chung Su* présente l'utilisation d'un préfiltre et d'un post-filtre pour éliminer les perturbations sur le signal de la commande et la sortie du système. Dans ce cas, le signal devient lisse, le temps retard peut être réduit [6]. *Krupp, D* présente l'utilisation d'un filtre pour fournir un effet de lissage sur le signal de la commande discontinu [7]. *Yong-Seok Kim* utilise d'un observateur de perturbation basée sur la théorie de commande binaire. Avec l'aide de l'observateur, les perturbations d'observation sont éliminées [8]. *Faa-Jeng Lin* et *Wen-Der Chou* présente une application de la commande par mode glissant avec les algorithmes génétiques. Ces algorithmes sont simples et adaptatifs qui peuvent d'ailleurs être utilisés pour estimer la limite des incertitudes et ensuite, réduire le temps retard [9]. *Edwards* présente deux nouveaux schémas numériques pour trouver la surface commutative optimale afin de construire la commande par mode glissant [10]. Il faut remarquer que tous les auteurs ont proposé des solutions dont la prémisse se base sur l'utilisation du plan de phase. Cependant, dans mon mémoire, on a

proposé la méthode de remplacement du plan de phase par le plan d'état pour construire une nouvelle commande en forme explicite, la commande glissante. Avec la prédiction des trajectoires dans le plan d'état, la commande glissante peut être appliquée sans retard.

Le chapitre 5 introduit le plan d'état et présente la transformation du plan de phase au plan d'état pour trouver la base de la commande glissante. Cette transformation est présentée dans deux situations : pôles réels et pôles complexes conjugués. Dans ce cas, la base de la commande glissante consiste en deux parties :

1. Un plan d'état réel trouvé en cas de pôles réels.
2. Une sphère de *Riemann* construite en cas de pôles complexes conjugués.

Le chapitre 6 présente la démonstration de l'application de la commande glissante. Grâce à la commande glissante qui est en forme explicite, celle-ci peut être appliquée sans retard. Avec cette particularité, on a trouvé que l'application de la commande glissante peut minimiser le temps de la réponse temporelle. Par conséquent, le remplacement du plan de phase par le plan d'état est nécessaire et intéressant.

Enfin, chaque chapitre présenté dans ce mémoire est étayé par d'exemples de simulation dans le logiciel *Matlab*. Pour cela, on construit la plate-forme des calculs d'approximation des réponses temporelles et les trois niveaux de la plate-forme informatique. Ces plates-formes sont présentées dans les annexes et utilisées pour illustrer des contenus écrits.

Chapitre 2 Modélisation des systèmes invariants dans le temps et l'application du modèle à paramètres variables

2.1 Introduction

De nos jours, la plupart des machines numériques et des robots sont considérés comme des systèmes complexes et généralement non linéaires. En industrie, la commande appliquée dans des systèmes a pour but d'assurer la sortie du système afin qu'il puisse atteindre la consigne désirée. La commande qu'on voudrait introduire dans ce mémoire est la commande glissante qui est construite à l'aide du plan d'état. Pour cela, la commande par mode glissant qui est construite dans le plan de phase est la base de la transformation pour trouver cette nouvelle commande. En effet, la commande par mode glissant est optimale puisqu'elle respecte le temps de réponse minimale et assure la stabilité du système. Il faut remarquer que la commande par mode glissant est une commande de structure variable. Alors, on choisit les familles de trajectoires en forme de col et en forme de foyer pour faire la construction de la commande par mode glissant dans le plan de phase. Par ailleurs, il faut définir une ligne droite comme la ligne commutative pour construire cette commande. L'équation de la ligne commutative est $LC = kx_1(t) + x_2(t)$. Tous les points appartenant à cette ligne sont exprimés comme $LC=0$. En observant le plan de phase (voir Fig. 2.1.1), il peut être divisé en 4 quadrants. Cependant, en supposant que la convergence des trajectoires est toujours garantie. Alors, la ligne commutative doit être placée dans le 2e et 4e quadrant (voir Fig. 2.1.1).

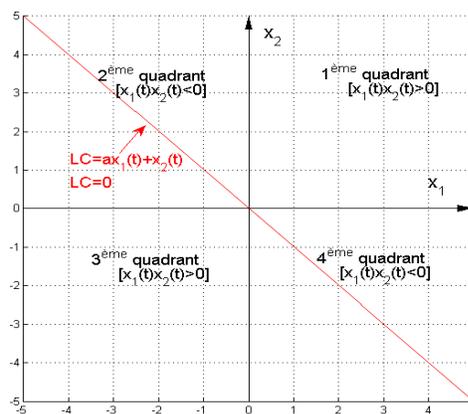


Figure 2.1.1 Ligne commutative et les quadrants du plan de phase

Selon la loi de la commande, les trajectoires du système changent sa forme lorsqu'elles se déplacent sur la ligne commutative. Ainsi, le système suivrait cette ligne en convergeant vers un point du plan de phase [11] (voir Fig. 2.1.2 a), courbe noire). Lorsque l'entrée du système est nulle, le point de convergence est le point de l'origine. Par ailleurs, on peut aisément présumer que le point de convergence du système en $x_2(t)$ est toujours zéro puisqu'il s'agit de la dérivée de $x_1(t)$ et que si $x_1(t)$ a atteint un régime stationnaire.

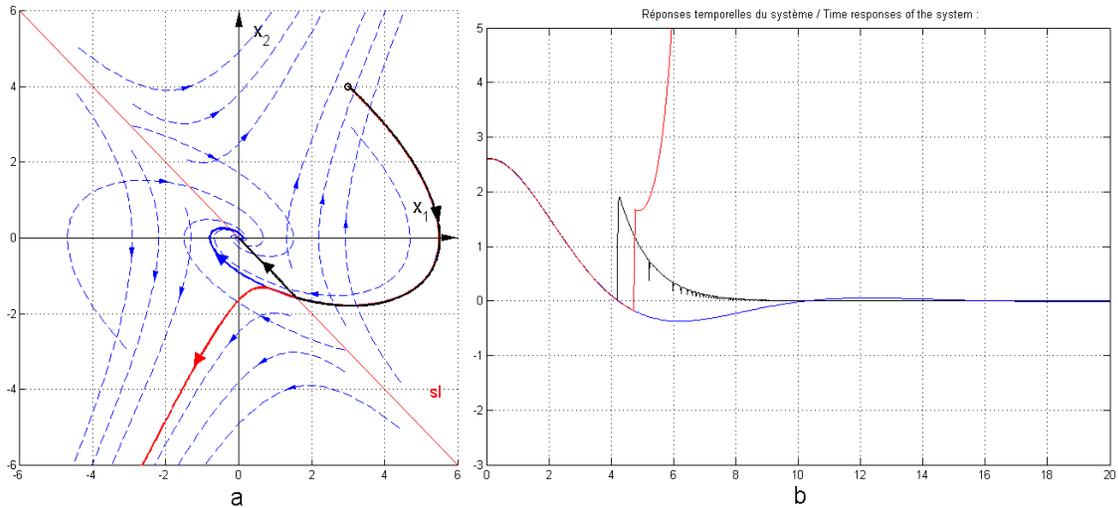


Figure 2.1.2 Exemple d'application de la commande par mode glissant idéal et réel, a) : Trajectoires du système dans le plan de phase, b) : Réponses temporelles du système

Alors, dans la figure 2.1.2, la courbe noire est la trajectoire d'application de la commande par mode glissant idéal avec l'entrée nulle. Sa ligne commutative est l'asymptote de col qui est placée dans le 2e et 4e quadrant. La courbe bleue est la trajectoire lorsque le système est en forme de foyer. La courbe rouge est la trajectoire d'application de la commande par mode glissant sous notre système étudié qui a un temps retard sur le changement des structures. Dans ce cas, la stabilité du système ne peut pas être garantie. Des discussions sur la commande par mode glissant seront présentées plus loin au chapitre 4 dans ce mémoire.

Pour appliquer la commande par mode glissant dans le plan de phase et la commande glissante dans le plan d'état, un modèle mathématique du système est requis. Pour cela, dans cette étude d'asservissement, une plate-forme mobile de soudage est utilisée pour faire voir les étapes de modélisation/identification et d'application des calculs numériques fréquentées en temps réel.

2.2 Modélisation/identification d'une plate-forme mobile de soudage

La modélisation ou l'identification d'un système est la conception d'un modèle pour représenter ce système. Elle repose sur les principes mathématiques et les lois physiques. Pour notre système étudié, la Plate-Forme Mobile (PFM) et son robot soudeur constituent un ensemble du système complexe. Par conséquent, il faut faire l'acquisition des données mesurées pour obtenir un modèle mathématique. Dans ce mémoire, le « *System identification Tools* » qui se trouve dans le logiciel *Matlab* est utilisé pour faire l'identification du système [12, 13, 14, 15, 16, 17]. Ainsi, la démonstration ci-dessous montre comment construire un modèle mathématique du système en forme d'une fonction de transfert.

Voici un exemple de construction d'un modèle mathématique de la PFM, et de l'acquisition des données afin de faire l'identification du système :

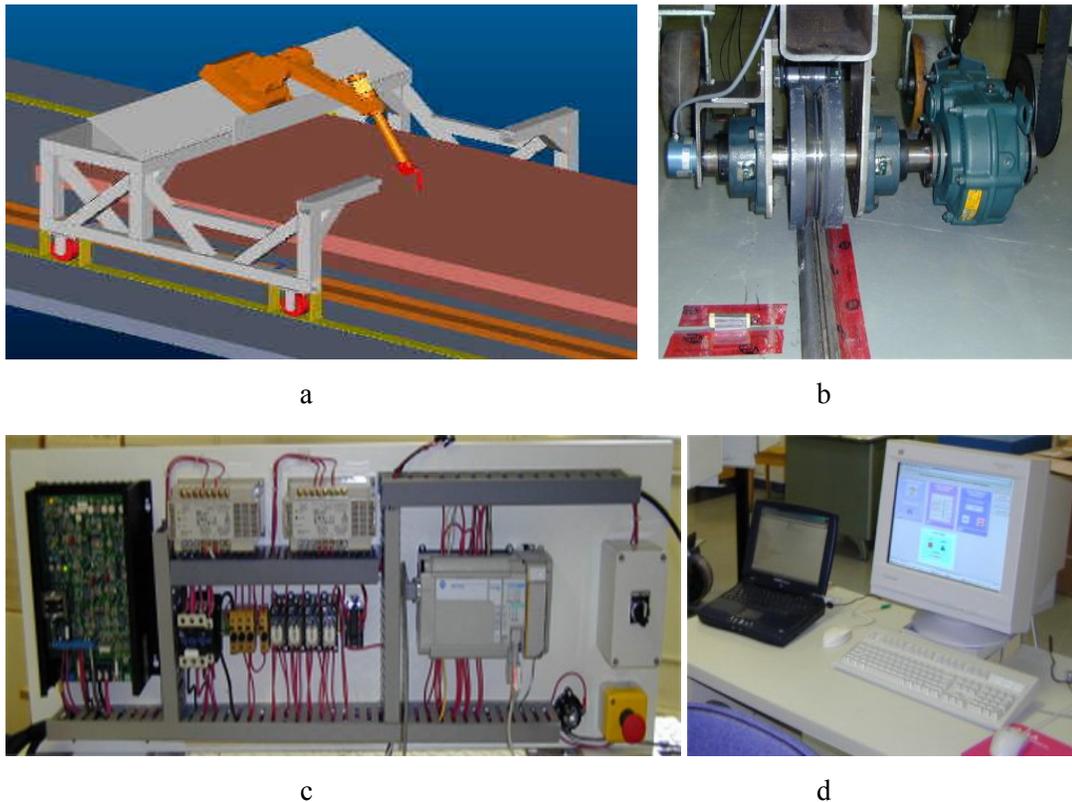


Figure 2.2.1 Reproduction d'un système complexe [12, 17] : « Système commandé de soudage à l'arc avec robot mobile », a) : « Plate-forme mobile avec quelques capteurs », b) : « Système de propulsion de la PFM », c) : « Panneau Électrique de la PFM », d) : « Réseau sur micro-ordinateur comme une optimisation du régulateur »

Il faut remarquer qu'un système dynamique est presque inévitablement soumis à plusieurs types de perturbations ou de bruits. Dans ce cas, on est confronté à deux cas possibles :

1. Le cas où l'influence du bruit est négligeable.
2. Le cas où l'influence du bruit est considérable.

Pour la première situation, lorsqu'on fait l'acquisition des données mesurées, l'entrée imposée au système est $u(t) = 1$. Et puis, une sortie en valeurs numériques est obtenue :

$$[0; 0,40; 0,63; 0,80; 0,93; 1,04; 1,12; 1,17; 1,22; 1,26; 1,29]$$

La réponse temporelle est reproduite par le logiciel *Simulink* est illustrée à la figure 2.2.2

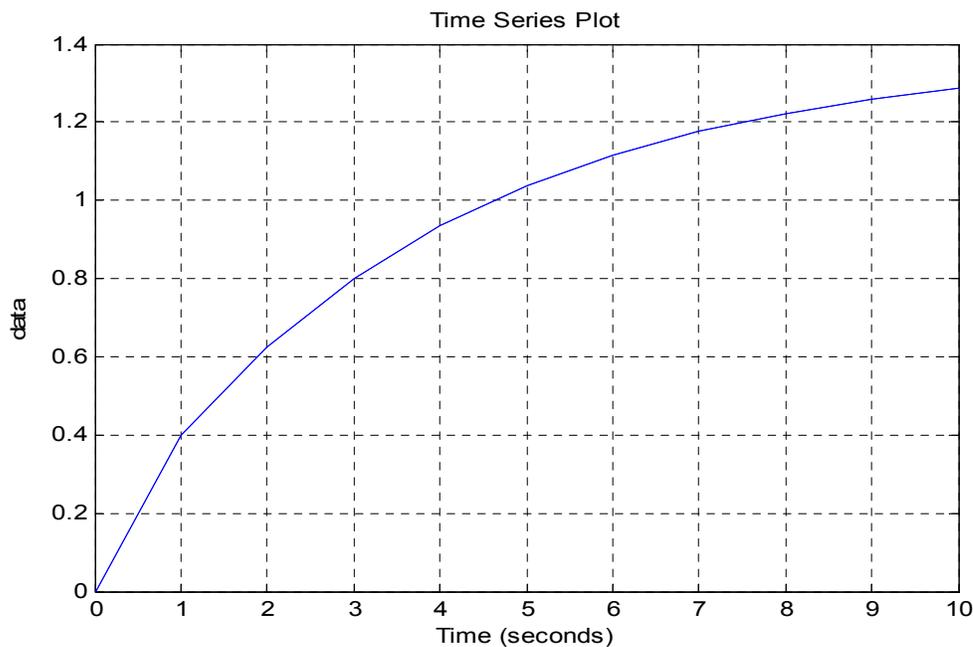


Figure 2.2.2 Simulation de la réponse temporelle sans perturbations

En utilisant le « *System identification Tools* », le modèle mathématique en forme d'une fonction de transfert ainsi obtenu. L'identification permet d'avoir l'expression de la fonction de transfert de la PFM à la forme suivante :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{s + 2,3}{s^2 + 6,6s + 1,67} \quad (2.1)$$

Pour la deuxième situation, la PFM est utilisée au Laboratoire avec des perturbations. Dans ce cas, lorsqu'on fait l'acquisition des données avec le même signal d'entrée, les valeurs numériques de sortie sont obtenues comme les valeurs suivantes :

$$[0; 0,5; 0,82; 1,1; 1,25; 1,35; 1,4; 1,41; 1,42; 1,42; 1,42]$$

$$[0; 0,5; 0,85; 1,2; 1,35; 1,45; 1,55; 1,6; 1,61; 1,62; 1,63]$$

Ensuite, les réponses temporelles en continu sont trouvées à la figure 2.2.3 :

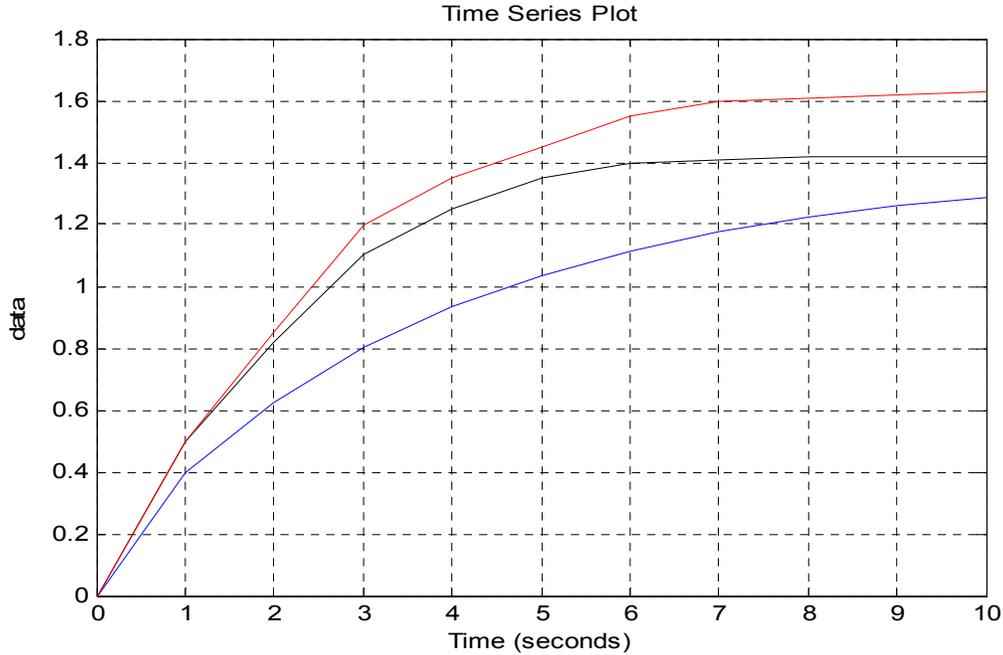


Figure 2.2.3 Simulation des réponses temporelles avec des perturbations

Les courbes rouge et noire sont des réponses temporelles qui sont trouvées avec des perturbations. Elles sont très différentes de la réponse temporelle sans perturbations (voir Fig.2.2.3 courbe bleue). Pour ces deux réponses temporelles avec perturbations, les modèles du système sont définis avec les formes suivantes :

Pour la réponse temporelle noire :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-1,41s^2 + 0,94s + 3,78}{s^4 + 6,07s^3 + 13,1s^2 + 9,38s + 2,69} \quad (2.2)$$

Pour la réponse temporelle rouge :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{7,33s^2 + 3,04s + 26,91}{s^4 + 9,97s^3 + 17,12s^2 + 41,82s + 16,2} \quad (2.3)$$

En raison de l'influence du bruit qui est considérable, le système devient plus complexe. Ainsi, on peut affirmer que la fonction de transfert du modèle du système soit représentée par un ordre supérieur que deux. Cependant, pour appliquer la commande par mode glissant et la commande glissante, il faut que la fonction de transfert du modèle de la PFM soit représentée par une fonction de transfert de second ordre. En effet, le modèle du second ordre est facile à analyser et faire le calcul numérique.

Alors, les modèles du système d'ordre supérieur doivent être réduits. Il existe plusieurs méthodes pour réduire l'ordre d'un modèle linéaire. Dans ce qui suit, on cite quelques méthodes de réduction classiques.

2.3 Méthodes de réduction classiques

Voici les méthodes de réduction classiques [18, 19, 20, 21, 22] :

1. Skogestad « règle de la moitié »;
2. Méthode de Padé;
3. Méthode des fractions continues;
4. Méthode d'approximation de Routh;
5. Méthode Schur;
6. Méthode d'« Optimal Hankel norme approximation ».

2.3.1 Skogestad « règle de la moitié »

Skogestad (2002) a proposé une méthode d'approximation pour les modèles d'ordre supérieur qui contiennent plusieurs constantes de temps. Il néglige la plus grande constante de temps à la façon suivante :

— La moitié de la valeur de la plus grande constante de temps est ajoutée au retard existant et l'autre moitié de celle est ajoutée à la plus petite constante de temps de retenue.

— Les constantes de temps qui sont plus petites que la plus grande constante de temps sont estimées comme des retards à usage.

Il pense qu'un retard peut être exprimé comme une expansion en série de Taylor :

$$e^{-\theta_0 s} \approx 1 - \theta_0 s \quad (2.4)$$

$$e^{-\theta_0 s} = \frac{1}{e^{\theta_0 s}} \approx \frac{1}{1 + \theta_0 s} \quad (2.5)$$

Donc, si la réduction de la fonction de transfert est de second ordre, la fonction de transfert peut être écrite la forme suivante :

$$G(s) = \frac{K e^{-\theta_0 s}}{(\tau_1 s + 1)(\tau_2 s + 1)} = \frac{K(1 + \theta_0 s)}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2.6)$$

où τ_1 est le plus grand retard au dénominateur, τ_2 et θ_0 sont trouvés par la règle de la moitié.

Il suffit de déterminer le temps retard, le gain statique et la constante de temps. Dans ce cas, le modèle réduit peut être construit.

2.3.2 Méthode de Padé

En mathématiques, l'approximant de Padé est une méthode d'approximation d'une fonction analytique par une fonction rationnelle. Donc, pour utiliser cette méthode, le système supérieur est défini comme :

$$G(s) = \frac{a_0 + a_1s + a_2s^2 + a_3s^3 + \dots + a_ms^m}{b_0 + b_1s + b_2s^2 + b_3s^3 + \dots + b_ns^n}; m < n \quad (2.7)$$

Et puis, le modèle d'ordre réduit est défini à la forme suivante :

$$R_k(s) = \frac{d_0 + d_1s + d_2s^2 + d_3s^3 + \dots + d_js^j}{e_0 + e_1s + e_2s^2 + e_3s^3 + \dots + e_ks^k}; j < k \& k < n \quad (2.8)$$

où $R_k(s)$ est le modèle de k ordre qui approche le système supérieur. Dans ce cas, avec l'utilisation de l'approximation de Padé, le système devient :

$$G(s) = c_0 + c_1s + c_2s^2 + c_3s^3 + \dots + c_ns^n$$

$$R_k(s) = r_0 + r_1s + r_2s^2 + r_3s^3 + \dots + r_ks^k$$

avec

$$r_i = c_i \quad (i = 0, 1, 2, 3 \dots k)$$

$$c_j = \frac{f^{(j)}(0)}{j!}; (j = 0, 1, 2, 3 \dots n)$$

$$f^{(0)}(0) = G(0)$$

$$f^{(1)}(0) = \dot{G}(0)$$

$$f^{(2)}(0) = \ddot{G}(0)$$

$$f^{(3)}(0) = \dddot{G}(0)$$

⋮

$$f^{(n)}(0) = \overset{n}{G}(0) \quad (2.9)$$

On définit que l' $e_0 = 1$ et ensuite trouve :

$$\left\{ \begin{array}{l} d_0 = c_0 \\ d_1 = c_1 + c_0 e_1 \\ d_2 = c_2 + c_0 e_1 + c_1 e_2 \\ \vdots \\ d_j = c_j + c_{j-1} e_1 + \dots + c_1 e_j + c_0 e_{j+1} \\ \vdots \\ 0 = c_{k-1} + c_{k-2} e_1 + \dots + c_1 e_{k+1} + c_0 e_k \end{array} \right. \quad (2.10)$$

Avec les équations ci-dessus, les valeurs de polynôme au numérateur et au dénominateur dans la fonction de transfert $R_k(s)$ peuvent être trouvées.

2.3.3 Méthode des fractions continues

En examinant la fonction de transfert en forme des fractions continues, on peut trouver qu'il existe les parties qui composent l'ordre supérieur. Ainsi, le modèle réduit peut être construit par l'élimination de ces parties. On considère à titre d'exemple pour présenter cette méthode. Dans ce cas, le système supérieur est défini comme :

$$G(s) = \frac{a_0 + a_1 s + a_2 s^2 + a_3 s^3 + \dots + a_m s^m}{b_0 + b_1 s + b_2 s^2 + b_3 s^3 + \dots + b_n s^n}; m < n \quad (2.11)$$

On peut le décomposer en forme des fractions continues :

$$G(s) = \frac{I}{a_1 + \frac{I}{a_2 \frac{I}{s} + \frac{I}{a_3 + \frac{I}{a_4 \frac{I}{s} \cdot a_{2n-1} + \frac{I}{a_{2n} \frac{I}{s}}}}} \quad (2.12)$$

Si on fait la réduction à l'ordre k, le modèle réduit en forme des fractions continues peut être écrit à la forme suivante :

$$R_k(s) = \frac{I}{a_1 + \frac{I}{a_2 \frac{I}{s} + \frac{I}{a_3 + \frac{I}{a_4 \frac{I}{s} \cdot a_{2k-1} + \frac{I}{a_{2k} \frac{I}{s}}}}} \quad (2.13)$$

Et ensuite, ce modèle réduit peut être écrit en forme normale :

$$R_k(s) = \frac{d_0 + d_1s + d_2s^2 + d_3s^3 + \dots + d_js^j}{e_0 + e_1s + e_2s^2 + e_3s^3 + \dots + e_ks^k}; j < k \text{ \& } k < n \quad (2.14)$$

Pour trouver les valeurs d' $a_i (i = 1, 2, 3, 4, \dots, k)$ dans la forme des fractions continues, il faut calculer la matrice de Routh. Dans ce cas, on change le système supérieur encore à la forme suivante pour faire le calcul :

$$G(s) = \frac{A_{21} + A_{22}s + A_{23}s^2 + A_{24}s^3 + \dots}{A_{11} + A_{12}s + A_{13}s^2 + A_{14}s^3 + \dots} \quad (2.15)$$

Alors, la matrice de Routh est trouvée :

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & \dots \\ A_{21} & A_{22} & A_{23} & A_{24} & \dots \\ A_{31} & A_{32} & A_{33} & A_{34} & \dots \\ A_{41} & A_{42} & A_{43} & A_{44} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (2.16)$$

Les éléments dans cette matrice satisfont aux équations ci-dessous :

$$A_{j,k} = A_{j-2,k+1} - A_{j-1,k+1} \frac{A_{j-2,1}}{A_{j-1,1}}; j \geq 3 \quad (2.17)$$

$$a_i = \frac{A_{i,1}}{A_{i+1,1}}; (i = 1, 2, 3, \dots, 2n)$$

Pour faire la transformation du modèle réduit en forme des fractions continues à la forme normale, il faut calculer une autre matrice :

$$\begin{bmatrix} B_{01} & 0 & \dots & 0 & 0 \\ B_{11} & 0 & \dots & 0 & 0 \\ B_{21} & B_{22} & \dots & 0 & 0 \\ B_{31} & B_{32} & \dots & 0 & 0 \\ B_{2k-1,1} & B_{2k-1,2} & \dots & B_{2k-1,k} & 0 \\ B_{2k,1} & B_{2k,2} & \dots & B_{2k,k} & B_{2k,k+1} \end{bmatrix} \quad (2.18)$$

Les éléments dans cette matrice satisfont aux équations suivantes :

$$\begin{cases} B_{01} = 1 \\ B_{i,1} = a_{2k+1-i} B_{i-1,1} (i = 1, 2, 3, 4 \dots 2k) \\ B_{0,p} = B_{1,p} = 0 (p > 1) \\ B_{i,p} = B_{i-2,p-1} + \frac{B_{i,1} B_{i-1,p}}{B_{i-1,1}} (i = 1, 2, 3, 4 \dots 2k; p > 1) \end{cases} \quad (2.19)$$

Les éléments dans deux dernières lignes de la matrice ci-dessus représentent les valeurs des coefficients des polynômes au numérateur et au dénominateur du modèle réduit.

2.3.4 Méthode d'approximation de Routh

Supposons maintenant un système d'ordre supérieur est décrit par la fonction de transfert à la forme suivante pour faire l'utilisation de la méthode de Routh :

$$G(s) = \frac{b_1 s^{n-1} + \dots + b_n}{a_0 s^n + a_1 s^{n-1} + \dots + a_n} \quad (2.20)$$

Ensuite, on fait une transformation :

$$G(s) = \beta_1 F_1(s) + \beta_2 F_1(s)F_2(s) + \dots + \beta_n F_1(s)F_2(s)\dots F_n(s)$$

$$F_i(s) = \frac{1}{\alpha_i s + \frac{1}{\alpha_{i+1} s + \frac{1}{\alpha_{i+2} s + \dots + \frac{1}{\alpha_{n-i} s + \frac{1}{\alpha_n s}}}}} \quad (2.21)$$

Dans ce cas, les tableaux de α et β doivent être trouvées pour faire l'utilisation de la méthode d'approximation de Routh.

Les tableaux de α et β peuvent être trouvés par les équations ci-dessous :

$$\begin{cases} a_0^{i+1} = a_2^{i-1} - \alpha_1 a_2^i \\ a_2^{i+1} = a_4^{i-1} - \alpha_i a_4^i \\ \vdots \\ a_{n-i-2}^{i+1} = a_{n-i}^{i-1} - \alpha_i a_{n-i}^i \end{cases} \quad (i = 1, 2, 3, 4 \dots n-1) \quad (2.22)$$

Lorsque $n-i$ est impair, la dernière équation est remplacée par :

$$\begin{cases} a_{n-i-1}^{i+1} = a_{n-i+1}^{i-1} \\ \alpha_i = \frac{a_0^{i-1}}{a_0^i} \\ \beta_i = \frac{b_0^i}{a_0^i}; (i = 1, 2, 3, 4 \dots, n) \\ b_{j-2}^{i+2} = b_j^i - \beta_i a_j^i; (i = 1, 2, 3, 4 \dots, n-2); j = \begin{cases} 2, 4, \dots, n-i \\ 2, 4, \dots, n-i-1 \end{cases} \end{cases} \quad (2.23)$$

$$\begin{cases}
B_1(s) = \beta_1 \\
A_2(s) = \alpha_1 \alpha_2 s^2 + \alpha_2 s + 1 \\
B_2(s) = \alpha_2 \beta_1 s + \beta_2 \\
A_3(s) = \alpha_1 \alpha_2 \alpha_3 s^3 + \alpha_2 \alpha_3 s^2 + (\alpha_1 + \alpha_3) s + 1 \\
B_3(s) = \alpha_2 \alpha_3 \beta_1 s^2 + \alpha_3 \beta_2 s + (\beta_1 + \beta_3) \\
\vdots \\
A_k = \alpha_k A_{k-1}(s) + A_{k-2}(s) \\
B_k(s) = \alpha_k B_{k-1}(s) + B_{k-2}(s) + \beta_k
\end{cases}$$

avec

$$k = 1, 2, 3 \dots \quad (2.25)$$

$$A_{-1}(s) = 0; B_{-1}(s) = 0$$

$$A_0(s) = 1; B_{-1}(s) = 0$$

Dans ce cas, il est possible de déterminer les coefficients du modèle réduit par les calculs des valeurs d'équations ci-dessus.

2.3.5 Méthode de Schur

Le logiciel **Matlab** permet d'obtenir le modèle d'ordre réduit à partir d'un système d'ordre supérieur selon des techniques différentes. Par exemple, la fonction de ***schmr()*** peut être utilisée pour réduire un système instable tandis que la fonction de ***modred ()*** peut être utilisée pour réduire un système stable. Ces deux techniques décomposent la fonction de transfert du système en deux parties : une partie contient les grandes valeurs singulières de la réponse en fréquence du système dynamique. Et l'autre partie contient les petites valeurs singulières de la réponse en fréquence du système dynamique. Les deux fonctions font une élimination de la partie dernière pour réduire l'ordre.

2.3.6 Méthode d'« Optimal Hankel norme approximation »

Dans le logiciel **Matlab**, le « **Control Tools Box** » possède une fonction ***ohklmr()*** qui est basée sur la méthode d'« Optimal Hankel norme approximation ». Cette technique peut être utilisée pour réduire l'ordre du système. Le modèle réduit trouvé par cette technique approxime bien de la réponse temporelle origine.

2.4 Exemple des réductions de l'ordre d'un modèle mathématique de la PFM subi d'une perturbation

En utilisant les méthodes proposées ci-dessus, on a obtenu les résultats suivants :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-1,41s^2 + 0,94s + 3,78}{s^4 + 6,07s^3 + 13,13s^2 + 9,38s + 2,69} \quad (2.26)$$

Le système étudié avec perturbation est choisie comme exemple pour faire la réduction. Comme les pôles du système sont complexes conjugués, la première méthode présentée dans le chapitre 2.3 ne peut pas être utilisée. Dans ce cas, les modèles réduits sont trouvés :

Par la méthode de Padé :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-0,27s + 0,51}{s^2 + 0,99s + 0,37}$$

Par la méthode des fractions continues :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-0,27s + 0,51}{s^2 + 0,99s + 0,37}$$

Par la méthode d'approximation de Routh :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{0,083s + 0,33}{s^2 + 0,82s + 0,24}$$

Par la méthode de Schur :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-0,26s + 0,49}{s^2 + 0,91s + 0,36}$$

Par la méthode d'« Optimal Hankel norme approximation » :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{-0,35s + 0,53}{s^2 + 1,0s + 0,39}$$

Les réponses temporelles de ces modèle réduits sont trouvées (voir Fig. 2.4.1) :

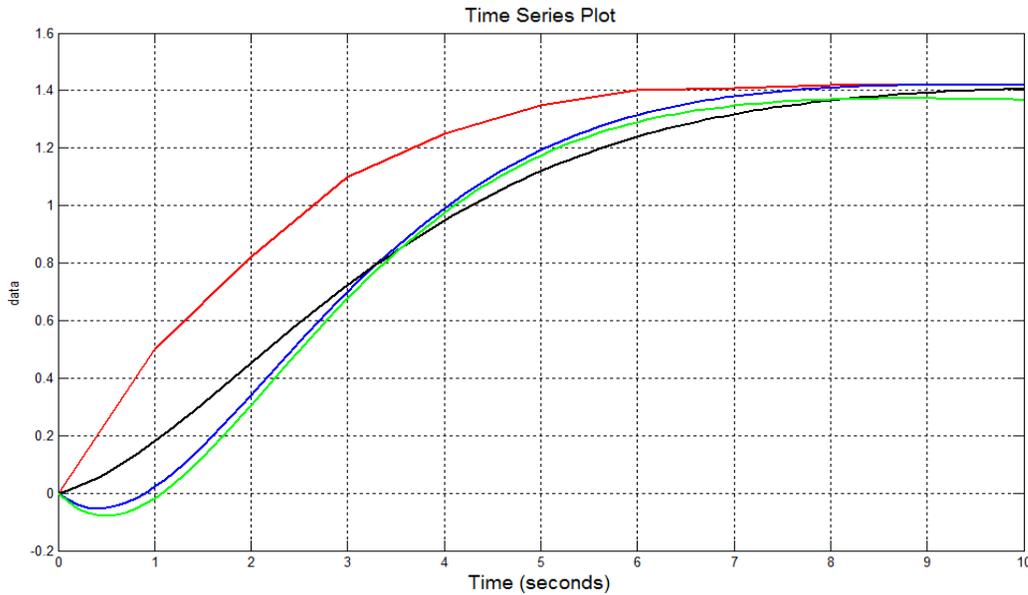


Figure 2.4.1 Réponses temporelles de la PFM

La courbe rouge est la réponse temporelle du système avec perturbations. Les autres courbes sont les réponses temporelles des modèles réduits. La démonstration précédente n'approxime pas correctement la réponse temporelle du système étudié (voire Fig. 2.4.1 courbe rouge). Ainsi, ces méthodes de réduction ne sont pas utilisables pour construire le modèle du système de second ordre. Par conséquent, on propose une fonction de transfert comme la forme suivante [12, 17, 23, 24, 25] :

$$G_{SL}(s) = \frac{G_{\theta}(s) + G_C(s)}{1 + G_{\theta}(s)G_C(s)} \quad (2.27)$$

avec :

$$G_{\theta}(s) = \frac{K}{1 + Ts} \quad (2.28)$$

$$G_C(s) = K_p + \frac{K_i}{s} \quad (2.29)$$

Où : $G_{\theta}(s)$ est la fonction de transfert d'ordre réduit au premier ordre, $G_C(s)$ est la fonction de transfert d'un correcteur PI. $G_{SL}(s)$ est la combinaison de deux ci-dessus en boucle de retour. Grâce au correcteur ajustable, ce modèle devient un modèle à paramètres variables (pour plus d'information, voir **Annexe A : Plate-forme des calculs d'approximation des réponses temporelles**). L'avantage d'adopter un modèle à paramètres variables, c'est qu'il donne la possibilité d'adapter les paramètres à l'évolution du système réel. Le calculateur numérique vérifie la validité du modèle en le comparant aux données fournies par le système réel. Dans ce cas, le modèle peut suivre correctement la réponse

du système étudié dans chaque incrément de temps. L'ajustement continu des paramètres du modèle garantit la validité du modèle.

Voici l'application de ce modèle ayant les mêmes conditions qu'un modèle subit des perturbations (voir Éqn. 2.26). La réponse temporelle du modèle peut être obtenue en ajustement des paramètres du correcteur :

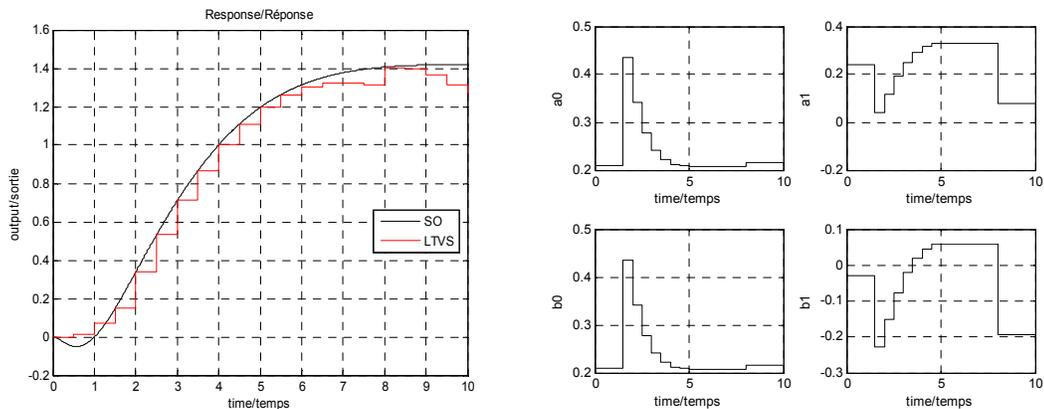


Figure 2.4.2 Démonstration de l'application du modèle à paramètres variables

La réponse temporelle du modèle en escalier reproduit une manière acceptable en présentant la réponse du système en cas d'une perturbation. Pour cela, le modèle est construit à la forme suivante :

$$G_0(s) = \frac{K}{I + T_S} = \frac{1,37}{I + 3,7s} = \frac{0,37}{s + 0,27}, \quad G_{SL}(s) = \frac{0,37(K_p s + K_i)}{s^2 + (0,37K_p + 0,27)s + 0,37K_i} \quad (2.30)$$

où les coefficients de $G_{SL}(s)$ sont :

$$\begin{aligned} a_0 &= 0,37K_i \\ a_1 &= 0,37K_p + 0,27 \\ b_0 &= 0,37K_i \\ b_1 &= 0,37K_p \end{aligned}$$

Généralement, pour une réponse temporelle donnée (voir Fig. 2.4.2 courbe noire), la courbe en escalier peut suivre correctement de cette réponse temporelle d'un système étudié. Grâce aux paramètres ajustables du correcteur (voir Fig. 2.4.2, et Éqn. 2.30), le modèle peut exprimer fidèlement la dynamique réelle du système. Alors, il est adaptatif. De plus, pour modéliser les systèmes qui sont plus complexes, le correcteur PI peut être remplacé par le correcteur PID. Par ailleurs, il est possible d'ajuster la fonction de transfert réduite dans chaque incrément de temps pour faire la modélisation correctement.

2.5 Conclusion

Ce chapitre a présenté la méthode de modélisation et l'identification pour construire un modèle mathématique qui peut être applicables de la commande par mode glissant et la commande glissante. Le modèle mathématique identifié et réduit à second ordre, sans perturbations et avec perturbations. Il faut noter ici qu'il existe de grandes erreurs entre le système d'ordre supérieur et les modèles réduits. Alors, on a proposé un modèle à paramètres variables comme une solution acceptable. La structure de ce modèle (voir Éqns. 2.27 à 2.29) contient un correcteur ajustable, une fonction de transfert réduite et une boucle de retour. Grâce aux paramètres ajustables du correcteur, le modèle proposé permet de représenter adéquatement les réponses temporelles du système en cas des perturbations (voir Fig.2.4.2 et Éqn. 2.30). D'après la construction de ce modèle, on peut maintenant réaliser la construction de la commande par mode glissant dans le plan de phase et la construction de la commande glissante dans le plan d'état. La présentation de la commande par mode glissant sera discutée au chapitre suivant.

Chapitre 3 Base de la commande par mode glissant à partir des équations d'état

3.1 Introduction

Dans le chapitre 2, on a présenté les étapes de modélisation/identification d'un système et construit un modèle à paramètres variables. Alors, la fonction de transfert du système étudié est exprimée comme :

$$G_{SL}(s) = \frac{0,37(K_p s + K_i)}{s^2 + (0,37K_p + 0,27)s + 0,37K_i} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \quad (3.0)$$

où les paramètres sont :

$$\begin{aligned} a_0 &= 0,37K_i \\ a_1 &= 0,37K_p + 0,27 \\ b_0 &= 0,37K_i \\ b_1 &= 0,37K_p \end{aligned} \quad (3.1)$$

Dans la majorité des cas, les équations d'état sont utilisées pour construire la base des commandes par modes glissants. Alors dans ce qui suit, la méthode de décomposition de la fonction de transfert en forme générale est présentée pour définir ces équations d'état.

3.2 Équations d'état en forme de phase

Généralement, un système est représenté par une fonction de transfert suivante [26, 27, 28] :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0}{s^{n+1} + a_n s^n + \dots + a_1 s + a_0} \quad (3.2)$$

Son schéma fonctionnel est construit à la manière suivante [12, 17, 29] :

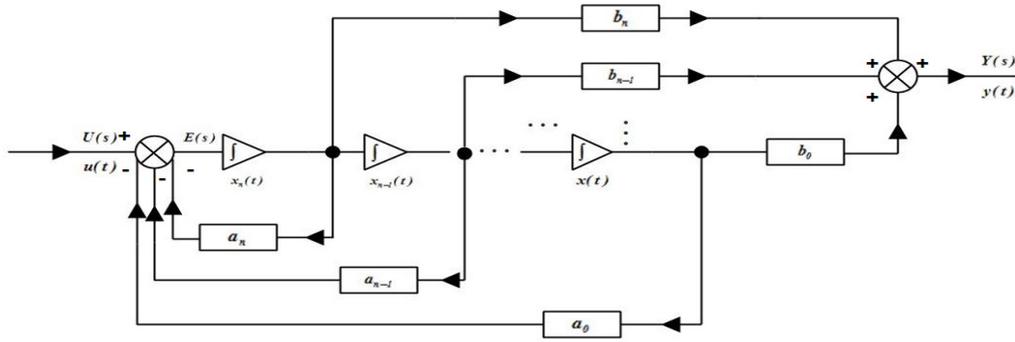


Figure 3.2.1 Schéma fonctionnel d'état

Alors, les variables d'état sont les sorties des intégrateurs, connectés en cascade, ce qu'expriment les équations d'état [17, 26, 28, 30] :

$$\begin{cases}
 \dot{x}_1(t) = \frac{dx_1(t)}{dt} = x_2(t) \\
 \dot{x}_2(t) = \frac{dx_2(t)}{dt} = x_3(t) \\
 \dots\dots\dots \\
 \dot{x}_{n-1}(t) = \frac{dx_{n-1}(t)}{dt} = x_n(t) \\
 \dot{x}_n(t) = \frac{dx_n(t)}{dt} = -a_0x_1(t) - a_1x_2(t) - \dots - a_{n-2}x_{n-1}(t) - a_{n-1}x_n(t) + u(t) \\
 y(t) = b_0x_1(t) + b_1x_2(t) + \dots + b_{n-1}x_n(t)
 \end{cases} \quad (3.3)$$

Maintenant, on peut exprimer l'équation de phase à la forme suivante :

$$\frac{dx_n(t)}{dx_{n-1}(t)} = f[x_n(t), x_{n-1}(t), x_n(0), x_{n-1}(0), t] \quad (3.4)$$

Pour résoudre cette équation et tracer les trajectoires dans le plan de phase, il faut construire un programme de calcul en utilisant la méthode des isoclines [30, 31, 32, 33]. Cette méthode sera présentée ci-dessous pour le cas particulier, c'est-à-dire simple.

3.3 Méthode des isoclines

Lorsqu'on utilise cette méthode, la première étape consiste à obtenir des champs directionnels qui sont constitués des isoclines trouvées par différentes pentes de tangentes aux trajectoires. Dans la deuxième étape, les trajectoires de phase sont trouvées à l'aide de ces isoclines.

Par conséquent, tout d'abord, il faut trouver l'équation d'isoclines. En général, les isoclines sont définies à partir de l'équation suivante :

$$\frac{dx_2(t)}{dx_1(t)} = f[x_2(t), x_1(t)] \quad (3.5)$$

Lorsqu'On définit les valeurs de la pente de tangente sous chaque isocline $\frac{dx_2(t)}{dx_1(t)}$ comme des constantes C , les isoclines peuvent être trouvées en résolvant l'équation :

$$f[x_2(t), x_1(t)] = c \quad (3.6)$$

Démonstration d'application de cette méthode pour notre système de second ordre :

$$\begin{cases} \frac{dx_1(t)}{dt} = x_2(t) \\ \frac{dx_2(t)}{dt} = -a_0x_1(t) - a_1x_2(t) + u(t) \end{cases} \quad (3.7)$$

où l'entrée est défini comme $u(t) = 0$ et les conditions initiales sont définies comme :

$$-\infty < x_1(0) < +\infty, -\infty < x_2(0) < +\infty.$$

Alors, l'équation des isoclines peut être trouvée à partir de la forme suivante :

$$\frac{dx_2(t)}{dx_1(t)} = \frac{-a_0x_1(t) - a_1x_2(t)}{x_2(t)} \quad (3.8)$$

La dérivée $\frac{dx_2(t)}{dx_1(t)}$ définit la pente de tangente sous chaque isocline. Ainsi, on remplace cette dérivée

par le paramètre ρ et on réécrit l'équation des isoclines qui devient :

$$x_2(t) = \frac{-a_0}{\rho + a_1} x_1(t) \quad (3.9)$$

D'une part, le paramètre $\rho \in (-\infty, +\infty)$ est un ensemble de constantes. Alors, en changeant la valeur de paramètre ρ , une infinité d'isoclines peut être trouvée dans le plan de phase. D'autre part, ce paramètre ρ définit la pente de tangente placée sous chaque isocline. Lorsque chacune des isoclines adjacentes se rapproche les uns des autres, les pentes sous chaque isocline définissent la direction et leurs tangentes aux trajectoires considérées comme des lignes droites courtes. Elles sont la solution des trajectoires rapprochées. Alors, lorsqu'on relie toutes ces lignes droites courtes, elles peuvent être considérées comme les trajectoires quasi continues dans le plan de phase.

Voici un exemple, choisissons les valeurs d' $a_0 = 1$ et $a_1 = 1$, l'équation des isoclines peut être réécrite à la forme suivante :

$$x_2(t) = \frac{-1}{\rho+1} x_1(t) \quad (3.10)$$

Pour trouver quelques lignes d'isoclines dans le plan de phase, on définit le paramètre ρ comme des valeurs numériques présentées dans le Tableau 3.3.1 :

Tableau 3.3.1 Valeurs numériques de ρ

ρ	-6,68	-3,75	-2,73	-2,19	-1,84	-1,58	-1,36	-1,18
	4,68	1,75	0,73	0,19	-0,16	-0,42	-0,64	-0,82

Alors, les valeurs de la pente d'isoclines peuvent être calculées et présentées dans le Tableau 3.3.2 :

Tableau 3.3.2 Valeurs numériques de la pente d'isoclines

$\frac{-1}{\rho+1}$	0,18	0,36	0,58	0,84	1,19	1,73	2,75	5,67
	-0,18	-0,36	-0,58	-0,84	-1,19	-1,73	-2,75	-5,67

On trace les isoclines dans le plan de phase et on place, sur chaque isocline, la pente de son tangent (voir Fig. 3.3.1 flèches noires) définie par le paramètre ρ . Finalement, on a montré qu'une réalisation approximative des trajectoires est obtenue (voir Fig. 3.3.1).

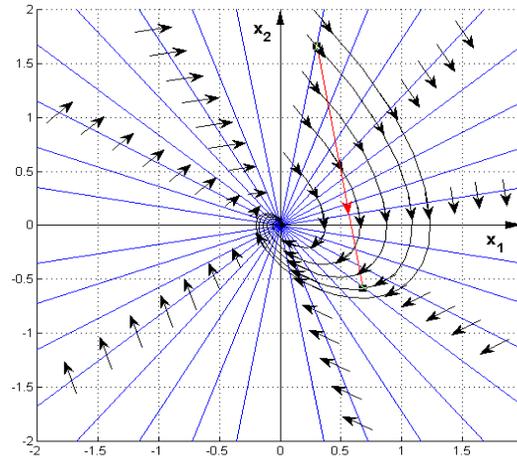


Figure 3.3.1 Trajectoires, en ligne brisée, sont trouvées

Il faut remarquer que dans la zone entre deux isoclines adjacentes, chaque trajectoire est remplacée par sa tangente. Alors on obtient finalement une trajectoire de phase en forme brisée, qu'on utilise particulièrement pour l'identification de la direction de cette trajectoire c'est-à-dire, la vérification de la stabilité du système [23, 24, 25, 28, 30, 31, 34]. Grâce à l'usage de calculs numériques qui relève de la technique du calcul analogique, la surface de cette zone se dirige vers zéro. Dans ce cas, la trajectoire calculée sera traitée comme une trajectoire quasi continue. Alors, on peut considérer que la précision de trajectoires calculées (voir Fig. 3.3.1 trajectoires noires et rouges) est définie par la minimisation de la surface de cette zone.

Maintenant, appliquant la notion des trajectoires de phase, pour l'équation 3.0 à l'aide des équations 3.3 et 3.4, on est conduit à la base de la commande par mode glissant.

3.4 Trajectoires de phase et leurs réponses temporelles

Avec l'ajustement des paramètres du correcteur dans le modèle (voir Éqn. 3.0), la méthode d'isoclines peut être utilisée pour construire les trajectoires dans le plan de phase. Ces trajectoires obtenues possèdent des propriétés du système dynamique en autre terme, la stabilité du système. À l'aide des

programmes construits par le logiciel *Matlab*, on a obtenu les trois points singuliers du système : col, foyer et nœud.

1) Trajectoires de phase en forme de col

En premier lieu, si les paramètres du correcteur sont :

$$a) : K_p = -4,78; K_i = -2,71,$$

$$b) : K_p = -3,44; K_i = -5,41,$$

$$c) : K_p = -0,73; K_i = -10,81,$$

$$d) : K_p = 34,41; K_i = -81,08.$$

Alors, on peut trouver les quatre familles de trajectoires de phase à la Figure 3.4.1 :

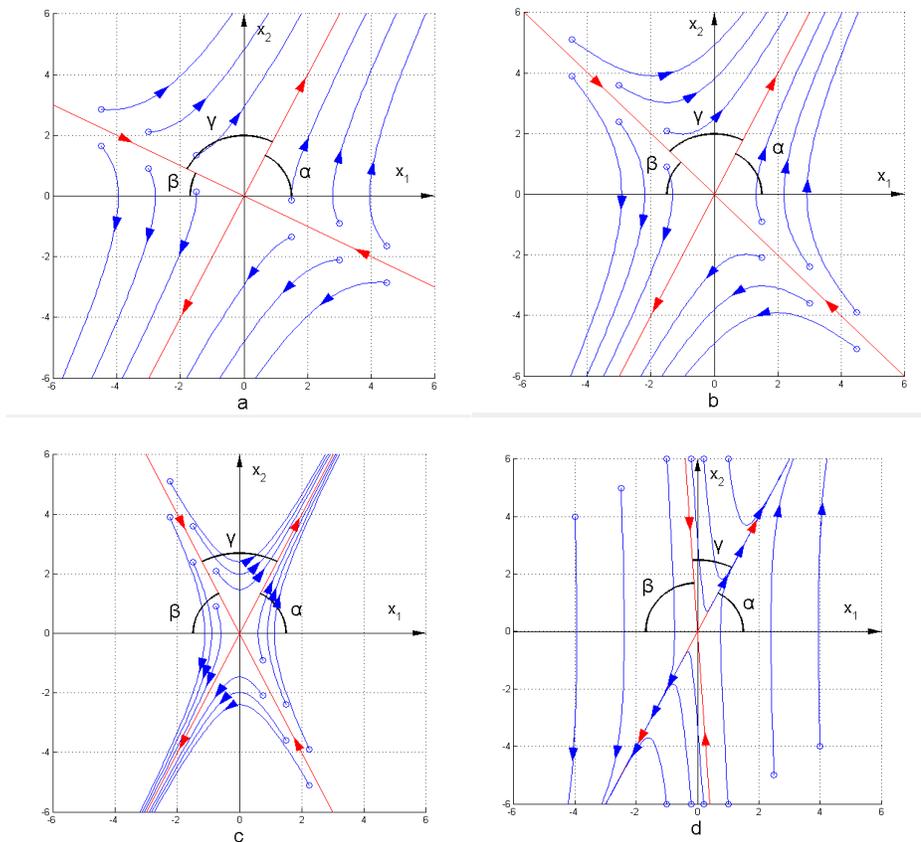


Figure 3.4.1 Familles de trajectoires de phase en forme de col (hyperbole)

On voit que les trajectoires de phase sont formées d'arcs des hyperboles et de deux asymptotes. Dans les quatre régions, composées par ses asymptotes, le système se comporte comme un système instable qui diverge à partir de n'importe quelle condition initiale. Cependant, à cause de l'existence d'asymptote qui se trouve dans les 2e et 4e quadrants, si une condition initiale se trouve sur cette asymptote le système se comporte comme un système stable.

On définit que les angles des asymptotes sont α et β respectifs. L'angle entre les deux asymptotes est γ (voir Fig. 3.4.1). Ces angles sont définis par des paramètres de la fonction de transfert du système (voir Éqn. 3.1) qui peuvent être calculés par l'équation suivante :

$$\begin{aligned} \tan(\alpha) &= \frac{-a_1 + \sqrt{a_1^2 - 4a_0}}{2} = \frac{-0,37K_p - 0,27 + \sqrt{(0,37K_p + 0,27)^2 - 1,48K_i}}{2} \\ \tan(\beta) &= \frac{-a_1 - \sqrt{a_1^2 - 4a_0}}{2} = \frac{-0,37K_p - 0,27 - \sqrt{(0,37K_p + 0,27)^2 - 1,48K_i}}{2} \end{aligned} \quad (3.11)$$

Alors, on trouve :

$$\begin{aligned} \alpha &= \left| \arctan \left[\frac{-0,37K_p - 0,27 + \sqrt{(0,37K_p + 0,27)^2 - 1,48K_i}}{2} \right] \right| \\ \beta &= \left| \arctan \left[\frac{-0,37K_p - 0,27 - \sqrt{(0,37K_p + 0,27)^2 - 1,48K_i}}{2} \right] \right| \\ \gamma &= \pi - (\alpha + \beta) \end{aligned} \quad (3.12)$$

Dans ce cas, les équations des asymptotes de quatre familles sont trouvées :

- a) : $x_2(t) = -0,5x_1(t)$, *stable*; $x_2(t) = 2x_1(t)$, *instable*
- b) : $x_2(t) = -x_1(t)$, *stable*; $x_2(t) = 2x_1(t)$, *instable*
- c) : $x_2(t) = -2x_1(t)$, *stable*; $x_2(t) = 2x_1(t)$, *instable*
- d) : $x_2(t) = -15x_1(t)$, *stable*; $x_2(t) = 2x_1(t)$, *instable*

Maintenant, on suppose [29] que ces asymptotes construiraient les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ si l'angle γ entre eux est : $\left(0 < \gamma < \frac{\pi}{2}\right)$ et $\left(\frac{\pi}{2} < \gamma < \pi\right)$ (voir Fig. 3.4.2 a) et b)).

D'où l'on déduit que ces coordonnées obliques sont les projections des coordonnées rectangulaires

$[v_1(t), v_2(t)]$ de plan d'état sur le plan de phase. Cela montre aussi l'existence d'angle $\gamma = \frac{\pi}{2}$ ayant les propriétés de l'énoncé, mais elles deviendront les coordonnées rectangulaires $[v_1(t), v_2(t)]$ (voir Fig. 3.4.2 c) et d)). Cette supposition est la base pour trouver les relations géométriques entre le plan de phase et le plan d'état lorsque les pôles du système sont réels.

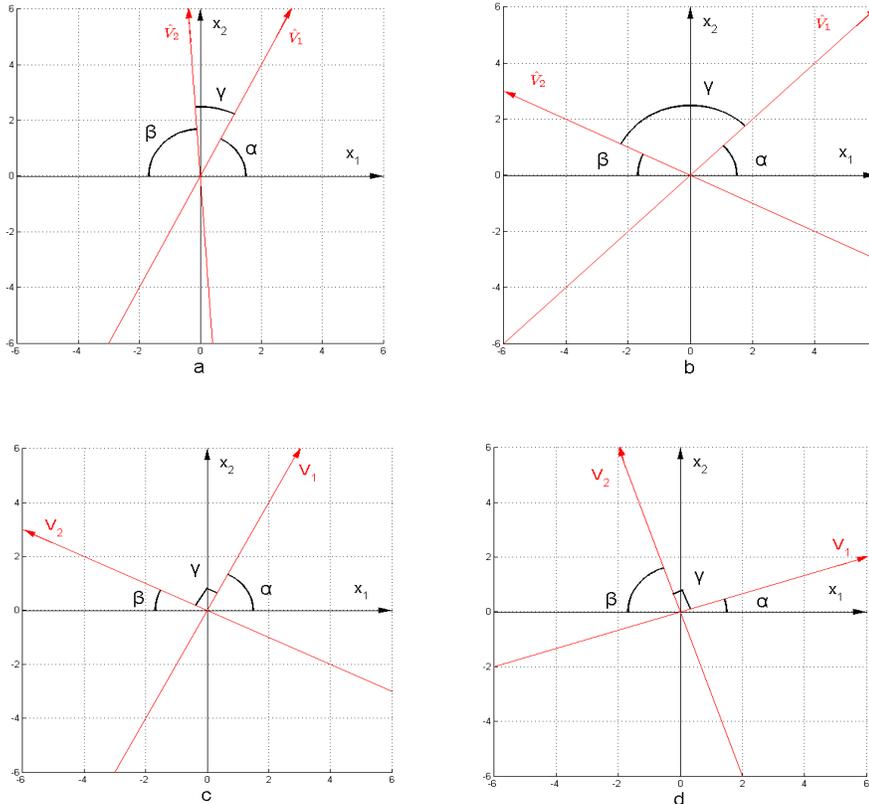


Figure 3.4.2 Constructions les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ et les coordonnées rectangulaires $[v_1(t), v_2(t)]$ dans le plan de phase, a) : $\gamma \in \left(0 < \gamma < \frac{\pi}{2}\right)$, b) : $\gamma \in \left(\frac{\pi}{2} < \gamma < \pi\right)$,

$$\text{c) et d) : } \gamma = \frac{\pi}{2}$$

Maintenant, si les conditions initiales se placent sur les asymptotes stables qui sont proposées ci-dessus, on obtient les réponses temporelles suivantes :

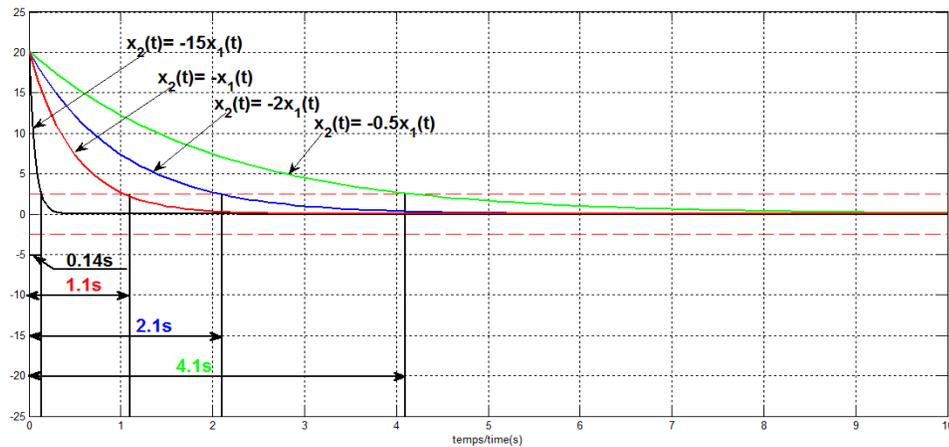


Figure 3.4.3 Réponses temporelles du système avec les conditions initiales se placent sur les asymptotes stables

Si le système atteint $\pm 10\%$ son régime définitif alors, on accepte sa réponse temporelle. On voit que pour les asymptotes choisies, les réponses temporelles se rapprochent de son régime définitif à $(0,14s; 1,1s; 2,1s; 4,1s)$. Dans ce cas, l'asymptote $x_2(t) = -15x_1(t)$ est la meilleur, mais en effet, l'asymptote le plus proche la coordonné $x_2(t) \rightarrow x_1(t) = 0$ est la meilleur. Dans ce cas, les réponses temporelles du système atteignent ses régimes définitifs rapidement. Alors, les asymptotes stables vont être utilisées plus loin au chapitre suivant pour construire la commande par mode glissant.

2) Trajectoires de phase en forme de foyer

En second lieu, si les paramètres du correcteur sont :

$$a) : K_p = 1; K_i = 1,$$

$$b) : K_p = 1; K_i = 3,$$

$$c) : K_p = -1; K_i = 1,$$

$$d) : K_p = -2; K_i = 3.$$

Les trajectoires de phase sont formées en spirale. Les quatre familles sont montrées à la figure 3.4.4 :

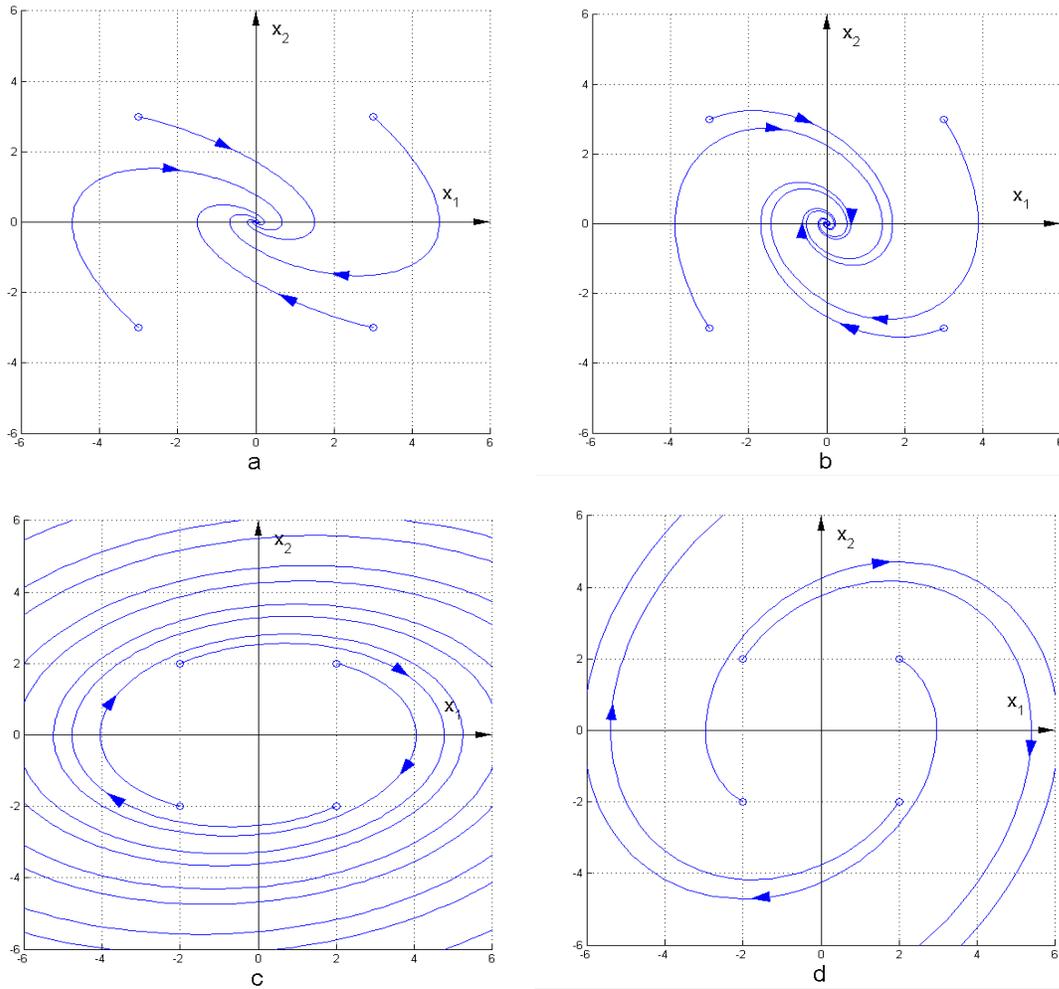


Figure 3.4.4 Familles de trajectoires de phase en forme de foyer (spirale)

Lorsque le système est représenté par les deux familles a) et b), il se comporte comme un système stable à partir de n'importe quelle condition initiale. Au contraire, si le système est représenté par les familles c) et d), ce système se comporte comme un système instable.

À la lumière de ce qui précède, si on choisit quelques conditions initiales dans la situation où la stabilité du système est garantie (voir Fig. 3.4.4 familles a) et b)). Alors, les réponses temporelles peuvent être trouvées à la figure 3.4.5 :

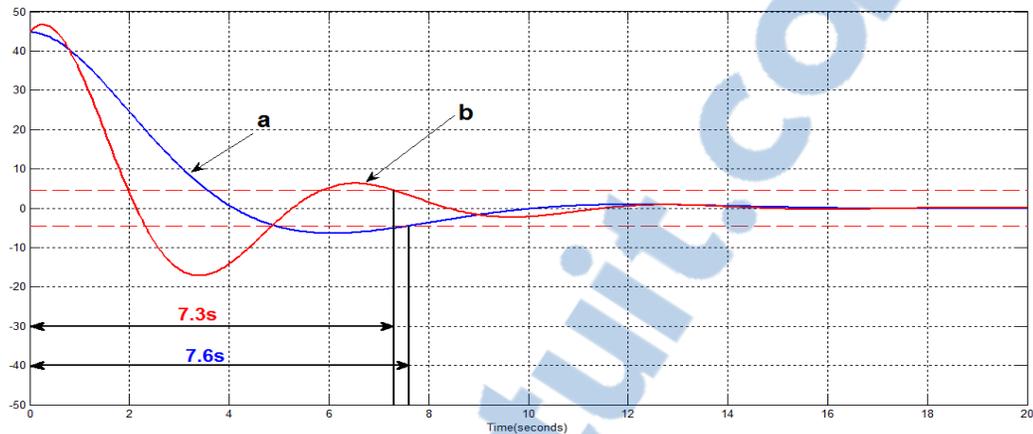


Figure 3.4.5 Réponses temporelles du système présenté en spirale

Selon la figure 3.4.5, les réponses temporelles sont en forme de sinusoïde et les réponses temporelles se rapprochent de son régime définitif à $(7,3s; 7,6s)$. Elles font l'oscillation avec long temps, mais la stabilité du système sera garantie. Par ailleurs, les réponses temporelles peuvent atteindre rapidement l'intervalle défini. Alors, les familles de trajectoires en forme de foyer vont être utilisées pour construire la commande par mode glissant.

3) Trajectoires de phase en forme de nœud

En dernier lieu, quand les paramètres du correcteur sont :

$$a) : K_p = 7,35; K_i = 5,4,$$

$$b) : K_p = 12,78; K_i = 16,28,$$

$$c) : K_p = 6,03; K_i = 2,70,$$

$$d) : K_p = -8,84; K_i = 5,41.$$

Les trajectoires de phase sont formées en nœud. Les quatre familles sont montrées à la figure 3.4.6 :

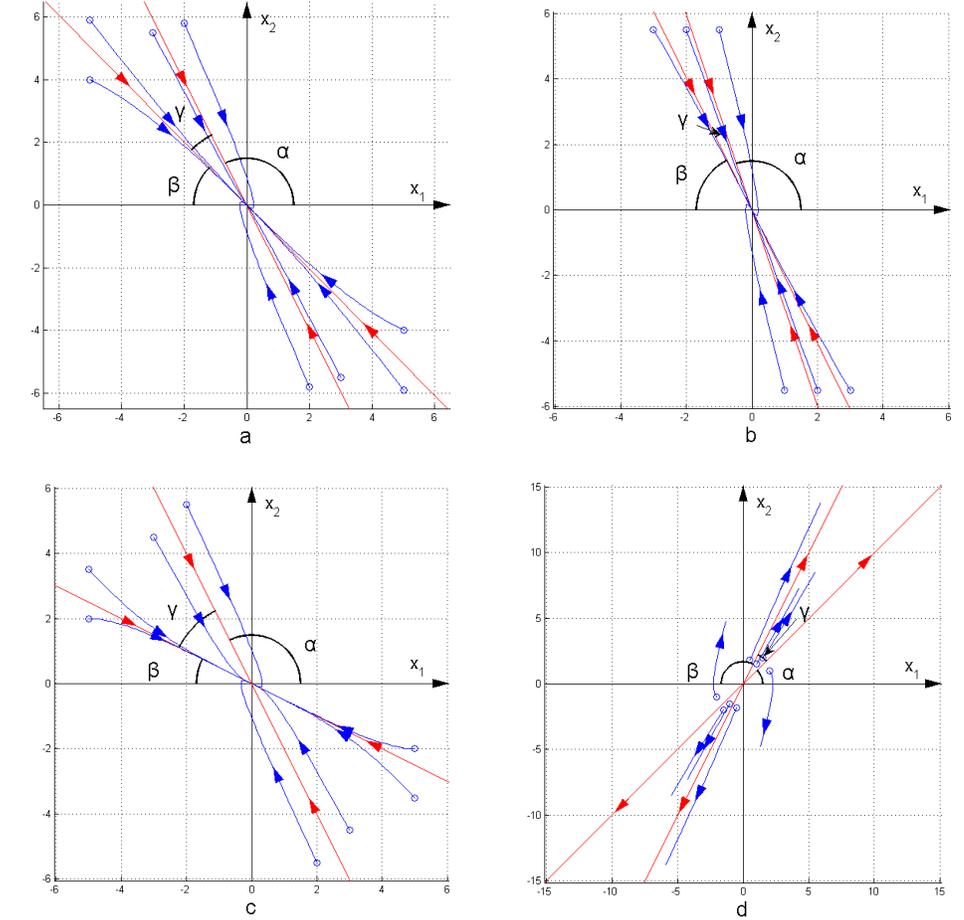


Figure 3.4.6 Familles de trajectoires de phase en forme de nœud

Si deux asymptotes se trouvent dans les 2e et 4e quadrants, le système est stable. Au contraire, si ces asymptotes se trouvent dans les 1ers et 3es quadrants le système est instable. Avec l'aide des équations 3.11 et 3.12, les équations de ces asymptotes peuvent être trouvées :

$$a) : x_2(t) = -x_1(t), \text{ stable}; x_2(t) = -2x_1(t), \text{ stable}$$

$$b) : x_2(t) = -3x_1(t), \text{ stable}; x_2(t) = -2x_1(t), \text{ stable}$$

$$c) : x_2(t) = -0,5x_1(t), \text{ stable}; x_2(t) = -2x_1(t), \text{ stable}$$

$$d) : x_2(t) = x_1(t), \text{ instable}; x_2(t) = 2x_1(t), \text{ instable}$$

Maintenant, en donnant les conditions initiales se placent sur l'asymptote stable, leurs réponses temporelles peuvent être trouvées :

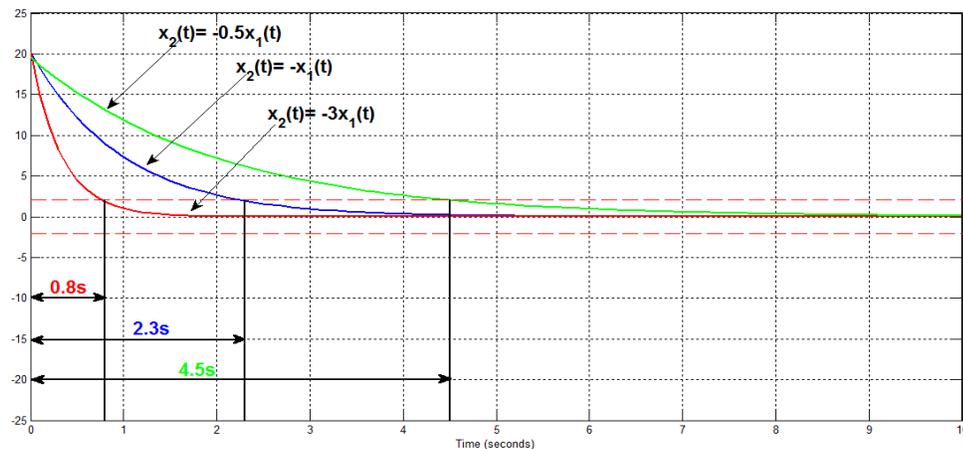


Figure 3.4.7 Réponses temporelles du système avec des conditions initiales d'asymptotes stables

La simulation, en absence de retard, montre lorsqu'on place les asymptotes dans le 2e et 4e quadrant de plan de phase, la condition initiale se stabilise plus rapidement qu'à la situation représentée par la figure 3.4.5 et ces quadrants peuvent devenir le siège de la stabilité. Cela signifie que le système doit rester dans ce siège [23, 24, 25, 28, 30, 32, 34].

Finalement, les deux familles de trajectoires de phase (voir Figs. 3.4.1 et 3.4.4) et le siège de la stabilité fournit le premier niveau de la base de la commande par mode glissant [11, 17, 34, 35, 36, 37, 38]. De même, la famille de trajectoire en forme de nœud (voir Fig. 3.4.6) peut remplacer la famille de trajectoire en forme de col (voir Fig. 3.4.1) pour construire de cette base. Le deuxième niveau de la base est la restitution du point initial de stabilité dans le plan de phase. Pour cela, l'observateur d'état est construit et utilisé.

3.5 Restitution des variables d'état avec un observateur

3.5.1 Introduction de l'observateur d'état général

L'observateur d'état [39, 40] permet de reconstruire les variables d'états à partir d'un modèle du système dynamique. Généralement, le modèle du système est présenté en forme des équations d'état comme la suivante :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ y(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \quad (3.13)$$

où $\mathbf{x}(t)$ désigne les variables d'états, $y(t)$ présente la réponse temporelle. Dans ce cas, la structure de l'observateur d'état peut être représentée sous la forme suivante :

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}e(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}y(t) \\ \hat{y}(t) = \mathbf{C}\hat{\mathbf{x}}(t) \end{cases} \quad (3.14)$$

L'observateur Luenberger est construit en temps continu, d'où $\hat{\mathbf{x}}(t)$ et $\hat{y}(t)$ sont les restitutions de variables d'état et de la réponse temporelle. Le terme correctif $\mathbf{L}e(t)$ fait apparaître comme le gain d'observateur relatif à l'erreur de la restitution de la réponse temporelle. L'erreur est présentée à la forme suivante :

$$\mathbf{e}(t) = y(t) - \hat{y}(t) \quad (3.15)$$

Dans le cas de l'application d'observateur d'état, on voudrait que l'erreur $\mathbf{e}(t)$ converge exponentiellement vers zéro :

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbf{e}(t) &= 0 \\ \lim_{t \rightarrow \infty} [y(t) - \hat{y}(t)] &= 0 \end{aligned} \quad (3.16)$$

Pour cela, on calcule l' $\dot{\mathbf{e}}(t)$:

$$\begin{aligned} \dot{\mathbf{e}}(t) &= \frac{d\mathbf{e}(t)}{dt} \\ &= \mathbf{C}[\dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t)] \\ &= \mathbf{C}[\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) - \mathbf{A}\hat{\mathbf{x}}(t) - \mathbf{B}\mathbf{u}(t) - \mathbf{L}e(t)] \\ &= (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}(t) \end{aligned} \quad (3.17)$$

Il est facile de trouver que la caractéristique dynamique d'erreur $\mathbf{e}(t)$ est définie par les valeurs propres de la matrice $(\mathbf{A} - \mathbf{L}\mathbf{C})$. Donc, il faut que ces valeurs propres soient toutes à parties réelles strictement négatives. Elles peuvent être calculées par l'équation suivante :

$$\det[\lambda \mathbf{I} - (\mathbf{A} - \mathbf{L}\mathbf{C})] = 0 \quad (3.18)$$

où λ ici présentent ces valeurs propres.

Par conséquent, le gain \mathbf{L} peut être obtenu par le choix des valeurs propres de la matrice $(\mathbf{A} - \mathbf{L}\mathbf{C})$. Ces valeurs propres présentent des pôles d'observateur d'état. Alors, la question d'ajustement du gain \mathbf{L} devient la question du placement des pôles d'observateur d'état. Généralement, on voudrait que la vitesse de la réponse temporelle restituée par l'observateur d'état est 2 à 5 fois plus rapide que celle du système. Alors, il faut que les parties réelles négatives des pôles d'observateur d'état soient plus petites que celles du système. Il faut remarquer que le choix des pôles d'observateur d'état n'est pas unique, plusieurs possibilités sont disponibles pour obtenir le gain \mathbf{L} . Ce gain doit être vérifié par la simulation.

Alors, lorsque les pôles d'observateur d'état ont été choisis, le gain \mathbf{L} peut être calculé par les fonctions *place()* ou *acker()* dans le logiciel *Matlab*. Par ailleurs, on peut utiliser la formule de Bass-Gura pour calculer ce gain \mathbf{L} . La formule de Bass-Gura est présentée suivante :

$$\mathbf{L} = (\mathbf{W}\mathbf{N}^T)^{-1} \begin{bmatrix} (\alpha_0 - a_0) \\ (\alpha_1 - a_1) \\ \vdots \\ (\alpha_{n-1} - a_{n-1}) \end{bmatrix} \quad (3.19)$$

où :

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} a_1 & \cdots & a_{n-1} & 1 \\ \vdots & & \ddots & 0 \\ a_{n-1} & \ddots & \cdots & \vdots \\ 1 & \cdots & \cdots & 0 \end{bmatrix} \\ \mathbf{N} &= \begin{bmatrix} \mathbf{C}^T & \mathbf{A}^T \mathbf{C}^T & \cdots & (\mathbf{A}^T)^{n-1} \mathbf{C}^T \end{bmatrix} \\ \det[s\mathbf{I} - \mathbf{A}] &= s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1s + a_0 \\ (s - \lambda_1) \cdots (s - \lambda_n) &= s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \cdots + \alpha_1s + \alpha_0 \\ n \in \mathfrak{R}, n &= 1, 2, 3, \cdots \end{aligned} \quad (3.20)$$

$\lambda_1, \cdots, \lambda_n$ Est le choix des pôles d'observateur d'état.

Il faut remarquer que le système est observable lorsqu'on utilise l'observateur d'état, le système doit être asymptotiquement stable. Dans ce cas, les valeurs propres de la matrice \mathbf{A} qui présentent les pôles du système sont à parties réelles strictement négatives.

3.5.2 Constriction de l'observateur d'état

Dans le cas de notre système étudié, à l'aide d'un système d'identification, on construit un modèle à paramètres variables. La fonction de transfert de ce système est présentée dans l'équation 3.0. Lorsque le système est stable, les pôles de celui-ci sont réels ou complexes conjugués aux parties réelles strictement négatives. Dans ce cas, il est observable. Alors on peut construire l'observation du système à la manière suivante (voir Fig. 3.5.1) :

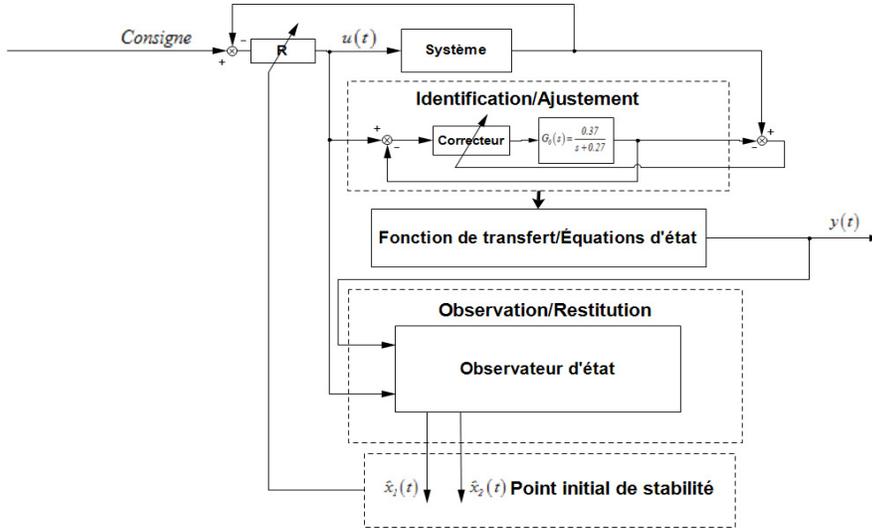


Figure 3.5.1 Schéma de l'observation

Il y a trois niveaux dans cette figure : identification, observation et restitution. Dans le premier niveau, on construit un modèle à paramètres variables et identifie à l'aide d'un système de l'acquisition des données mesurées (voir Chap. 2). Ainsi, le système étudié peut être présenté sous forme d'une fonction de transfert et sous forme des équations d'état.

Maintenant, on présente le système étudié à la forme d'équations d'état :

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -0,37K_i & -(0,37K_p + 0,27) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \mathbf{u}(t) \\ y(t) = [0,37K_i \quad 0,37K_p] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \end{cases} \quad (3.21)$$

où $A = \begin{bmatrix} 0 & I \\ -0,37K_i & -(0,37K_p + 0,27) \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$, $C = [0,37K_i \quad 0,37K_p]$

Avec l'aide de l'équation 3.14, l'observateur d'état de notre système peut être construit à la forme suivante :

$$\begin{bmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \end{bmatrix} = A \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} + Bu(t) + L \left\{ y(t) - C \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} \right\} \quad (3.22)$$

ou bien

$$\begin{bmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \end{bmatrix} = (A - LC) \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{bmatrix} + Bu(t) + Ly(t) \quad (3.23)$$

Finalement, le schéma fonctionnel du modèle et le schéma fonctionnel de l'observateur sont branchés en parallèle :

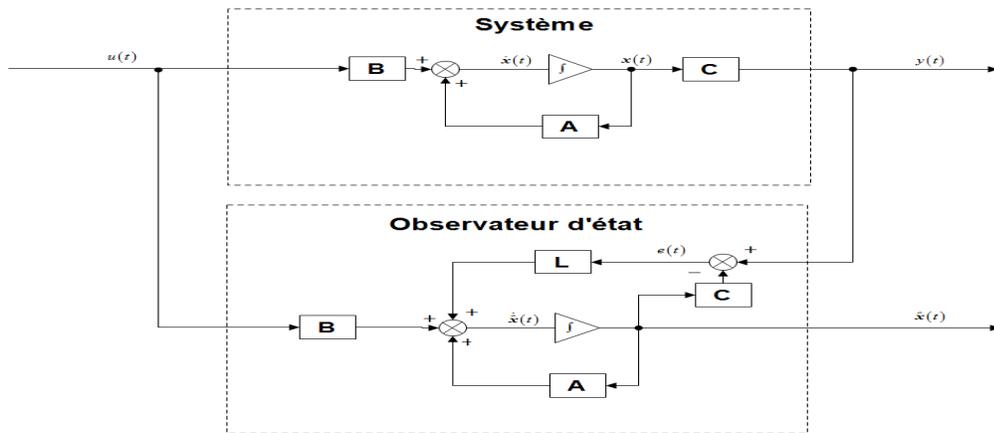


Figure 3.5.2 Schémas de l'observateur d'état présenté en forme d'équation 3.22

Et puis :

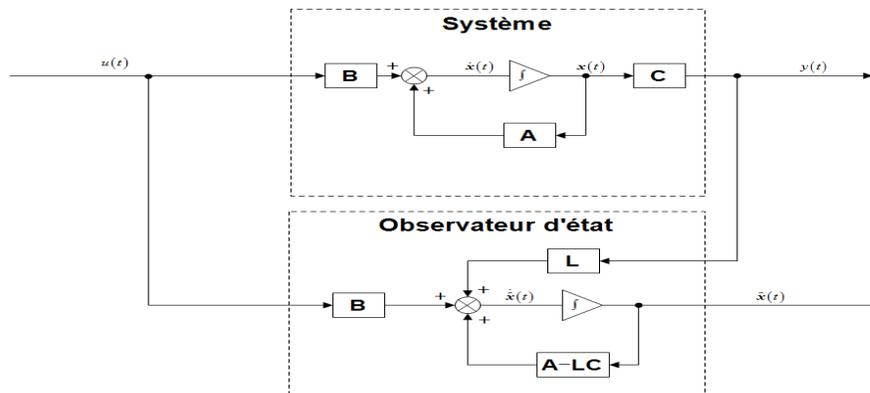


Figure 3.5.3 Schémas de l'observateur d'état présenté en forme d'équation 3.23

On trouve que les sorties d'observateur d'état $\hat{x}(t)$ sont les restitutions des variables d'état.

Voici l'exemple d'application de l'observateur d'état. Lorsque le système est stable, les paramètres du correcteur dans le modèle sont ajustés sur les valeurs suivantes :

$$K_p = 1; K_i = 3.$$

Dans ce cas, les pôles du système sont :

$$s_1 = -0,32 - j$$

$$s_2 = -0,32 + j$$

Ensuite, les valeurs propres de la matrice $(A - LC)$ sont définis :

$$\lambda_1 = -4,48 - j$$

$$\lambda_2 = -4,48 + j$$

Alors, l'erreur $e(t)$ va diminuer rapidement. Le gain d'observateur peut être obtenu à l'utilisation de la fonction de *Matlab place()* :

$$L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} 1,65 \\ 17,54 \end{bmatrix}$$

En même temps, on peut utiliser la formule de Bass-Gura pour calculer le gain L :

$$L = (WN^T)^{-1} \begin{bmatrix} (\alpha_0 - a_0) \\ (\alpha_1 - a_1) \\ \vdots \\ (\alpha_{n-1} - a_{n-1}) \end{bmatrix} = \left(\begin{bmatrix} 0,64 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1,11 & 0,37 \\ -0,41 & 0,87 \end{bmatrix} \right) \begin{bmatrix} (21,07 - 1,11) \\ (8,96 - 0,64) \end{bmatrix} = \begin{bmatrix} 1,65 \\ 17,54 \end{bmatrix}$$

Les gains d'observateur d'état obtenus par les deux façons sont les mêmes. Alors, à l'aide du logiciel *Simulink*, le schéma de l'observateur d'état peut être construit à la manière suivante (voir Fig. 3.5.4) :

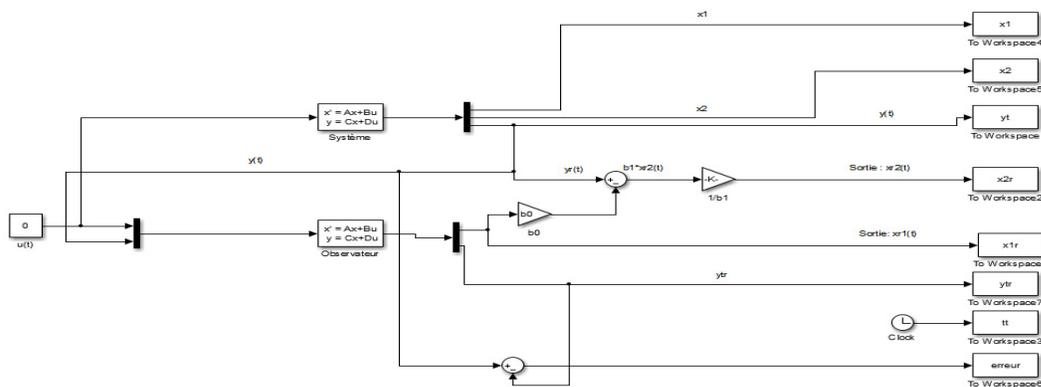


Figure 3.5.4 Schéma d'observateur d'état dans le *Simulink*

Et puis, la réponse temporelle mesurée et sa restitution sont montrées à la figure 3.5.5 :

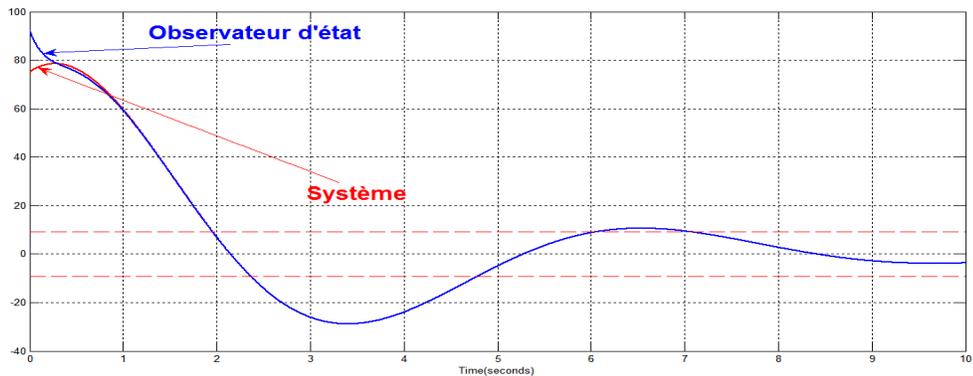


Figure 3.5.5 Réponses temporelles mesurées et restituées par l'observateur

Au début, les deux réponses temporelles sont différentes à cause d'erreur $e(t)$ est existée. Cette erreur est présentée à la figure 3.5.6 :

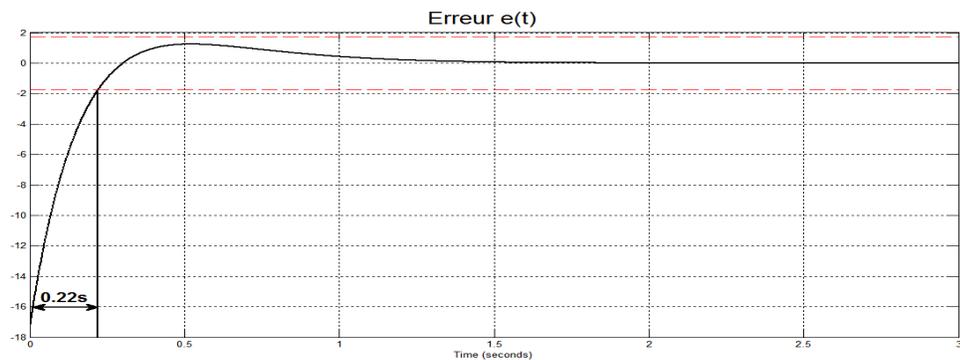


Figure 3.5.6 Erreur $e(t)$ en fonction du temps

Cette erreur diminue en fonction du temps. Si elle atteint $\pm 10\%$ les valeurs de leurs régimes définitifs. Alors, on accepte et considère que les variables d'état restituées par l'observateur d'état sont celles du modèle (système) qui est d'ailleurs aussi le temps retard $\tau = 0,22s$ dans l'exemple ci-dessus.

Dans notre observateur d'état, la restitution de variable d'état $\hat{x}_1(t)$ peut être mesurée directement :

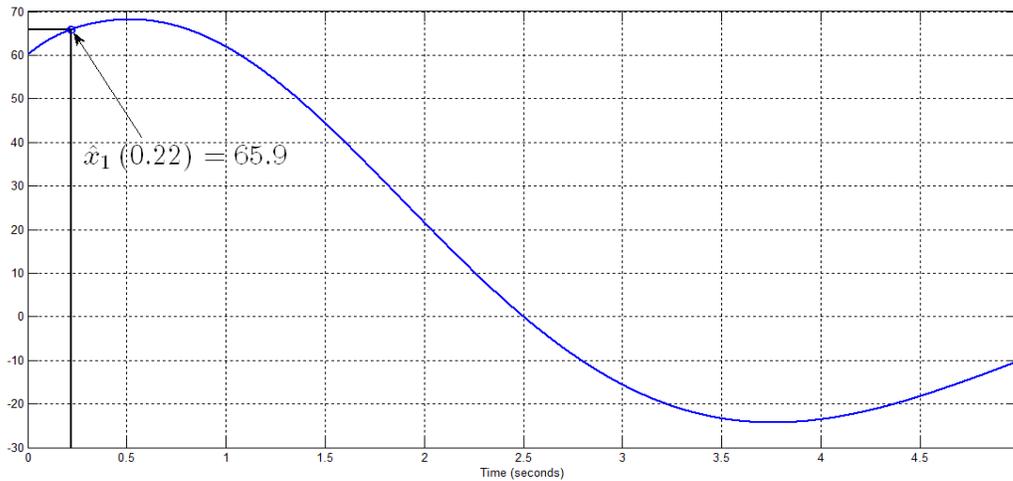


Figure 3.5.7 Restitution de variable d'état $\hat{x}_1(t)$

Après le temps retard $\tau = 0,22s$, on trouve la variable d'état $\hat{x}_1(0,22) = 65,9$.

La restitution de la variable d'état $\hat{x}_2(t)$ est calculée à l'aide de la réponse temporelle du système :

$$y(t) = 1,11\hat{x}_1(t) + 0,37\hat{x}_2(t) \quad (3.24)$$

En même temps, cette variable est la dérivée de la variable d'état $\hat{x}_1(t)$. Alors, il existe deux façons pour trouver celui-ci :

$$\hat{x}_2(0,22) = \frac{y(0,22) - 1,11\hat{x}_1(0,22)}{0,37} = \frac{78,43 - 73,16}{0,37} = 14,24$$

$$\hat{x}_2(0,22) = \frac{d\hat{x}_1(0,22)}{dt} = 16,3$$

On peut trouver qu'il existe une erreur entre les deux méthodes de calcul. Cette erreur est appelée l'erreur de restitution.

Maintenant, à l'aide de l'observateur d'état, les restitutions des variables d'état sont trouvées. Elles présentent le point initial de stabilité du modèle (système) dans le plan de phase :

$$\begin{bmatrix} \hat{x}_1(0,22) \\ \hat{x}_2(0,22) \end{bmatrix} = \begin{bmatrix} 65,9 \\ 14,24 \end{bmatrix}$$

Il faut remarquer que si le temps retard τ existe, la restitution de ce point est différente que celui du notre modèle (système). Il peut être décalé dans le plan de phase. Au contraire, si le temps retard τ

est extrêmement petit, la restitution de point initial de stabilité est la même que celui de notre modèle (système).

Par exemple, à cause du temps retard $\tau = 0,22s$, la restitution de point initial de stabilité est :

$$\begin{aligned}\hat{x}_1(0,22) &= 65,9 \neq x_1(0) \\ \hat{x}_2(0,22) &= 14,24 \neq x_2(0)\end{aligned}$$

Cependant, si le temps retard $\tau = 0s$, la restitution de ce point peut être trouvée comme la forme suivante :

$$\begin{aligned}\hat{x}_1(0) &= 50 = x_1(0) \\ \hat{x}_2(0) &= 53 = x_2(0)\end{aligned}$$

Maintenant, on place ces deux points dans le plan de phase à la figure 3.5.8 :

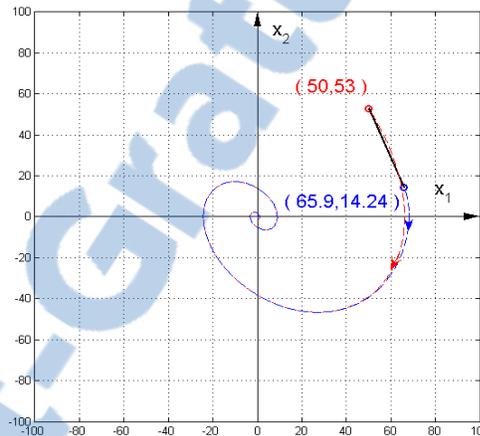


Figure 3.5.8 Distance du temps retard

La distance entre les deux points peut être calculée par l'équation suivante :

$$L_d = \sqrt{[\hat{x}_1(0) - \hat{x}_1(0,22)]^2 + [\hat{x}_2(0) - \hat{x}_2(0,22)]^2} = \sqrt{(-15,9)^2 + 38,76^2} = 41,9$$

Alors, on peut trouver que le temps retard fait un décalage du point initial de stabilité dans le plan de phase.

Finalement, avec les restitutions des variables d'état, le deuxième niveau de la base de la commande par mode glissant est trouvé. À l'aide de cette base, on peut construire et appliquer la commande par mode glissante.

3.6 Conclusion

On prépare la base de la commande par mode glissant qui consiste dans les familles de trajectoires de phase et la restitution du point initial de stabilité. Avec l'aide de cette base, la démonstration de la commande par mode glissante pourra être donnée au chapitre suivant en comparaison des résultats de la commande d'un processus idéalisé avec des résultats de restitution des variables d'état, cela exigée que chaque trajectoire parcourue par le système est se livrer par un temps retard τ . Ce temps retard est causé par la restitution du point initial de stabilité dans l'observateur d'état. Il va donner une grande influence lorsqu'on applique la commande par mode glissant dans le système étudié.

Chapitre 4 Commande par mode glissant et leur réponses temporelles dans le plan de phase

4.1 Introduction

Maintenant, à l'aide de la base de la commande par mode glissant, on va discuter de l'application de cette commande sous la base du plan de phase [4, 5, 6, 11, 35, 41, 42, 43]. En vue d'obtenir les informations d'application de la commande par mode glissant, une plate-forme informatique à trois niveaux est construite. La démonstration de l'application de cette commande est présentée dans le premier niveau de la plate-forme informatique. Une description détaillée de ce niveau de la plate-forme est à l'Annexe B : Plate-forme des calculs des commandes par modes glissants.

4.2 Vérifier la stabilité du système et le point initial de stabilité

À présent, supposons que l'entrée est constante. Lorsqu'on restitue le point initial de stabilité dans le plan de phase, notre système peut subir des perturbations, c'est-à-dire que la trajectoire du système est formée de col. Dans ce cas, le système devient instable (voir Fig. 4.2.2 trajectoire bleue). Par conséquent, il faut vérifier si les valeurs des paramètres du régulateur dans le système sont les mêmes que celles du correcteur dans notre modèle pour assurer la stabilité du système (voir Fig. 4.2.1). Si les valeurs ne sont pas les mêmes, il faut ajuster ces paramètres aux mêmes valeurs que ceux du correcteur.

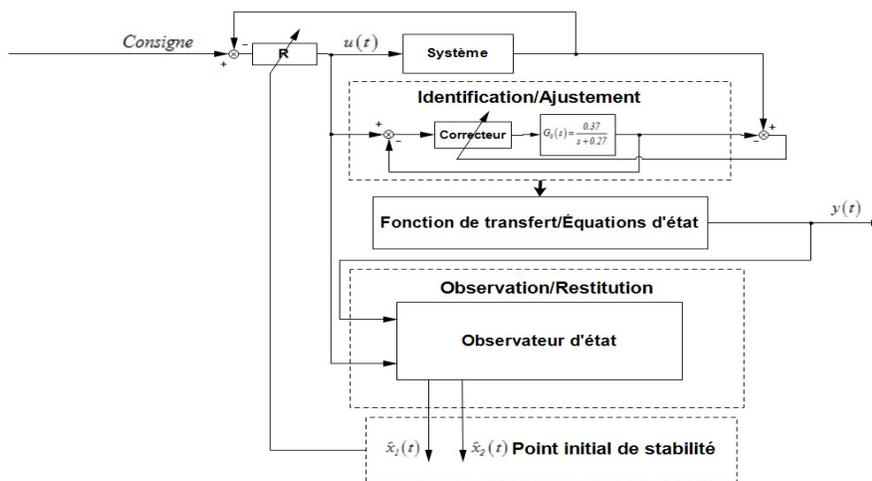


Figure 4.2.1 Schéma de l'observation

Lorsque le système est stable, on estime le point initial de stabilité dans le plan de phase.

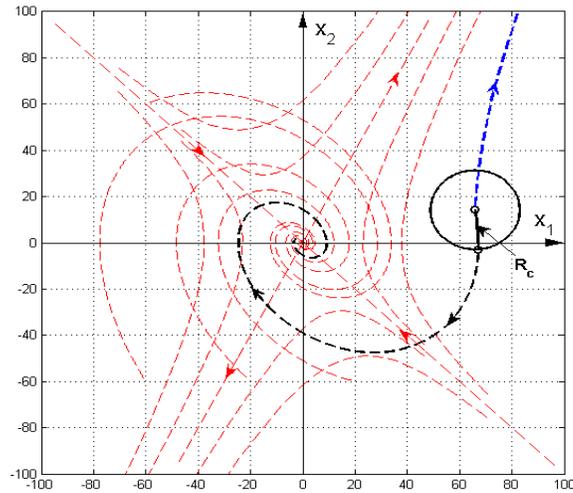


Figure 4.2.2 Vérification le point initial de stabilité

S'il n'existe pas de temps retard, le point initial de stabilité du modèle (système) est le même que le point estimé. Cependant, s'il existe le temps retard τ , ce point peut être décalé. Dans ce cas, le point initial de stabilité se trouve dans un cercle où le centre est le point estime, le rayon R_c est défini par le temps retard τ .

4.3 Commande par mode glissant d'un processus idéal

Supposons maintenant que le système est stable, le temps retard n'existe pas. Par ailleurs, on définit les paramètres du correcteur aux mêmes valeurs que l'exemple au chapitre 3.5 :

$$K_p = 1, K_i = 3.$$

La restitution du point initial de stabilité du modèle (système) dans le plan de phase est définie comme la forme suivante :

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 65,9 \\ 14,24 \end{bmatrix}$$

Dans ce cas, la trajectoire du système dans le plan de phase est formée de spirale (voir Fig.4.3.1) :

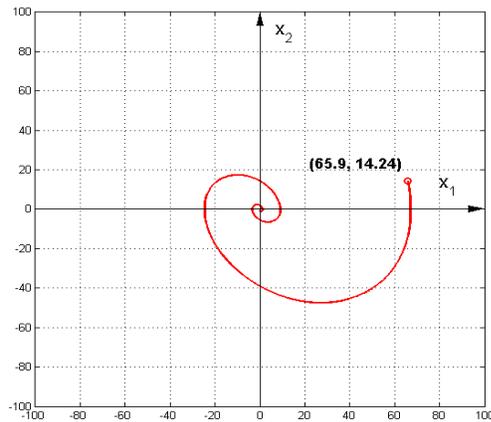


Figure 4.3.1 Trajectoire du système en forme de spirale (foyer)

Sa réponse temporelle est montrée dans la figure 4.3.2 :

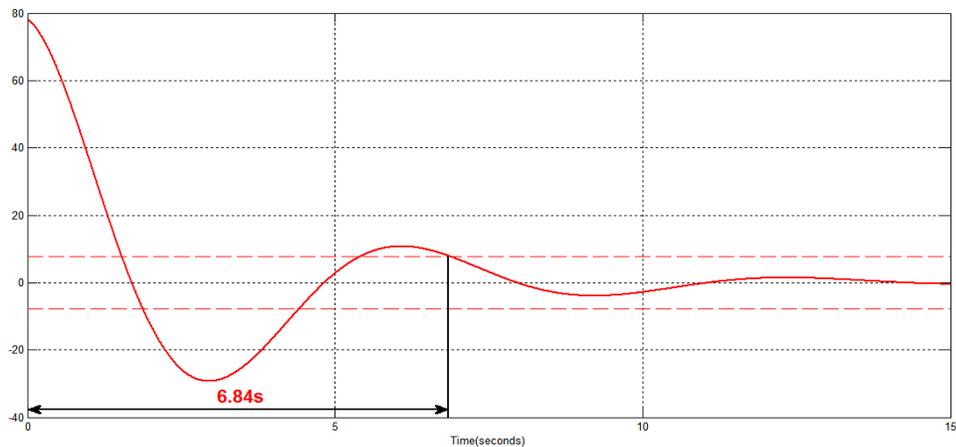


Figure 4.3.2 Réponse temporelle du système sous forme de spirale (foyer)

Selon la figure 4.3.2, lorsque le système atteint $\pm 10\%$ de son régime définitif, le temps de la réponse temporelle est $t_r = 6,84s$. Le système est stable, mais prend beaucoup du temps pour converger au point d'origine. Par conséquent, pour diminuer le temps de la réponse temporelle, la commande par mode glissant est appliquée. Grâce à l'asymptote stable qui est une trajectoire droite, on pourrait assurer la stabilité du système et en même temps, minimiser le temps.

Maintenant, en choisissant les asymptotes stables comme les suivantes, on peut construire quelques commandes par modes glissants :

$$x_2(t) = -0,5x_1(t); K_p = -4,78, K_i = -2,70,$$

$$x_2(t) = -x_1(t); K_p = -3,44, K_i = -5,41,$$

$$x_2(t) = -2x_1(t); K_p = -0,73, K_i = -10,81.$$

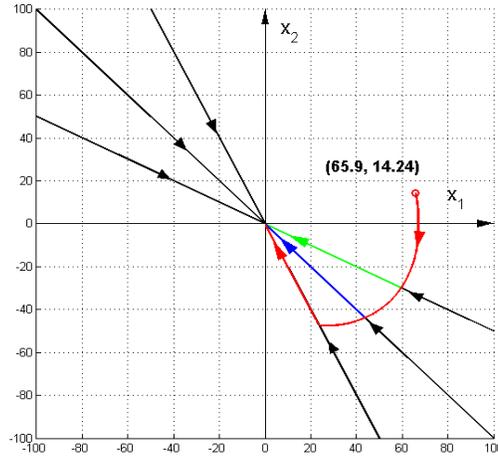


Figure 4.3.3 Trajectoires des commandes par modes glissants idéales

Lorsque les trajectoires se placent sur les asymptotes stables, on ajuste les paramètres du régulateur du système. Alors, le système change sa structure et suit ces asymptotes pour converger vers le point d'origine. Dans ce cas, on obtient des commandes par modes glissants d'un processus idéal. Ces asymptotes stables sont donc définies comme des lignes commutatives. Avec l'aide du premier niveau de la plate-forme informatique, les réponses temporelles sont trouvées :

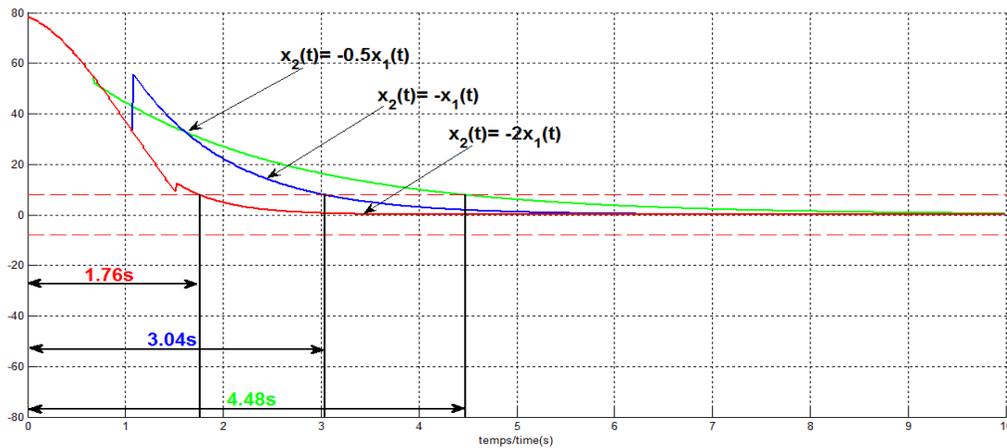


Figure 4.3.4 Réponses temporelles des commandes par modes glissants d'un processus idéal

Alors, il faut définir la ligne $x_1(t) = 0$ comme la deuxième ligne commutative pour faire la protection de la stabilité du système (voir Fig. 4.4.2).

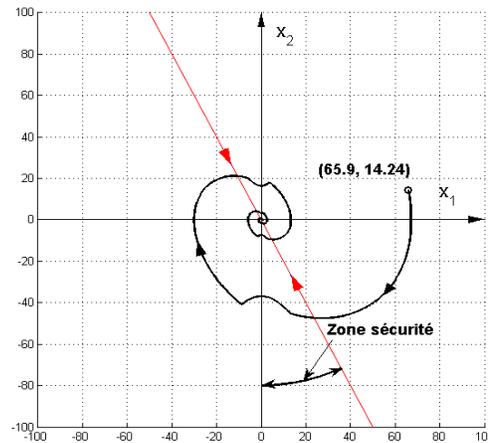


Figure 4.4.2 Trajectoire de la commande construite par deux lignes commutatives

Alors, les trajectoires de la commande sont des trajectoires spirales et des trajectoires d'hyperboles. Le système se dirige vers le point d'origine. Dans ce cas, il devient stable.

Lorsque le temps retard τ existe, il faut réserver une zone sécurité (voir Fig. 4.4.2) pour choisir l'asymptote stable. C'est une partie intéressante, mais n'est pas la question de cette recherche. Alors, dans notre exemple, en posant le temps retard $\tau = 0,22s$, la meilleure asymptote stable choisie est $x_2(t) = -2x_1(t)$.

Une fois que la stabilité est assurée, la réponse temporelle est trouvée (voir Fig. 4.4.3) :

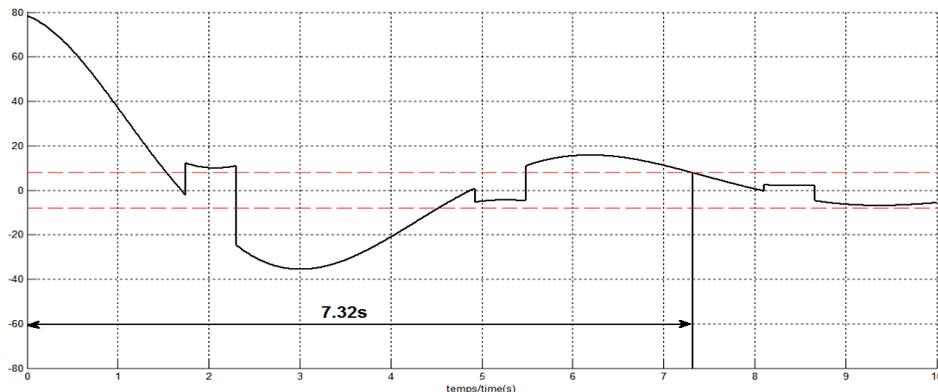


Figure 4.4.3 Réponse temporelle de la commande ci-dessus

Selon la figure 4.4.3, la réponse temporelle du système $t_r = 7,32s$. Il est plus long que la commande d'un processus idéal et il est aussi plus long que la valeur lorsque la trajectoire est formée de spirale (voir Fig. 4.3.2). Bien que la stabilité du système peut être garantie, mais le temps ne peut pas être diminué. Par conséquent, il faut définir la ligne commutative différente de l'asymptote stable lorsque le temps retard existe.

4.5 Commande par mode glissant au système étudié

Maintenant, plaçons une ligne droite entre la coordonnée $x_1(t) \rightarrow x_2(t) = 0$ et l'asymptote stable. (voir Fig. 4.5.1) :

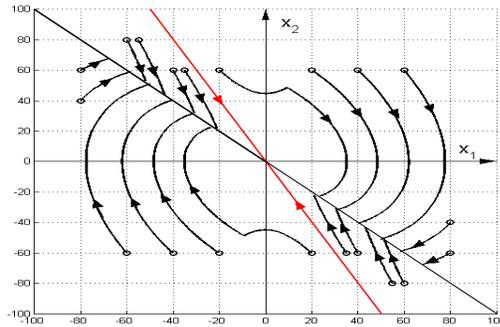


Figure 4.5.1 Définition la ligne commutative

À la surface, la stabilité du système est protégée par la définition de la ligne $x_1(t) = 0$ comme une ligne commutative. Pour cela, lorsque les conditions initiales se placent entre l'asymptote stable (voir Fig. 4.5.1 ligne rouge) et la ligne $x_1(t) = 0$, les trajectoires sont d'hyperboles. Cependant, si les trajectoires traversent la ligne $x_1(t) = 0$, les trajectoires sont de forme spirale. Cette situation est prédite selon le livre d'*Itkis* [4] et l'article d'*Utkin* [5]. Les conditions initiales qui se placent entre la ligne droite (voir Fig. 4.5.1 ligne noire) et l'asymptote stable ont des trajectoires qui sont formées des hyperboles. Elles se dirigent vers la ligne droite. Si les conditions initiales se placent dans les autres lieux sur la surface, les trajectoires sont formées des spirales. Dans le cas contraire où les trajectoires sont de forme spirale, ces dernières se dirigent aussi vers la ligne droite. Alors, on peut conclure que toutes les trajectoires convergentes vers cette ligne droite qui peut être définie comme la ligne commutative.

Voici l'exemple où la ligne commutative est définie comme $x_2(t) = -x_1(t)$:

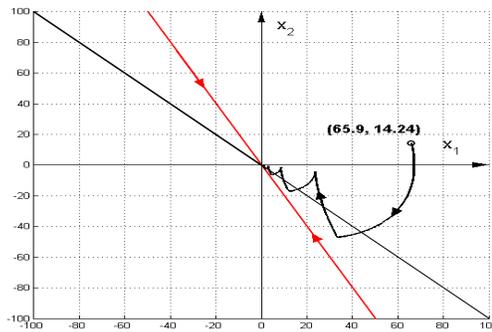


Figure 4.5.2 Trajectoire d'une commande par mode glissant

La trajectoire de cette commande est construite par la trajectoire spirale et la trajectoire de zigzag. Dans ce cas, le système est stable. Il se glisse autour de la ligne commutative au point d'origine. Finalement, on définit la forme de la commande par mode glissant à notre système étudié.

Avec l'aide du premier niveau de la plate-forme informatique, sa réponse temporelle peut être trouvée (voir Fig. 4.5.3) :

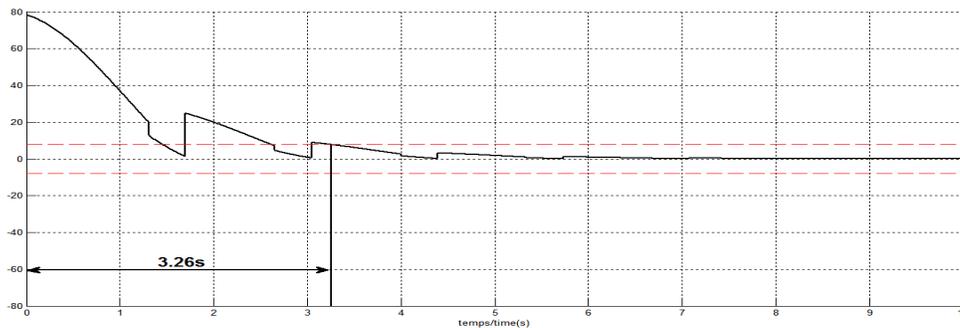


Figure 4.5.3 Réponse temporelle d'une commande par mode glissant

La réponse temporelle est $t_r = 3,26s$. Dans ce cas, le temps de réponse temporelle est amélioré.

Assurément, il existe une limite qui se rapproche de l'asymptote stable pour placer la ligne commutative où on assure en tout temps la stabilité et la contrôlabilité de notre système. Cette limite est influencée par le temps retard τ . De plus, toujours dans la mesure de réduire le temps de réponse temporelle, on peut être trouvé une autre limite plus proche de la coordonnée $x_1(t) \rightarrow x_2(t) = 0$.

Alors, en posant le temps retard $\tau = 0,22s$, les deux limites de déplacement de la ligne commutative sont trouvées :

$$x_2(t) = -1,35x_1(t) \text{ (Limite rapprochée à l'asymptote stable)}$$

$$x_2(t) = -0,6x_1(t) \text{ (Limite rapprochée à la coordonnée } x_1(t) \rightarrow x_2(t) = 0 \text{)}$$

Pour cela, un intervalle de déplacement de la ligne commutative est trouvé (voir Fig. 4.5.4) :

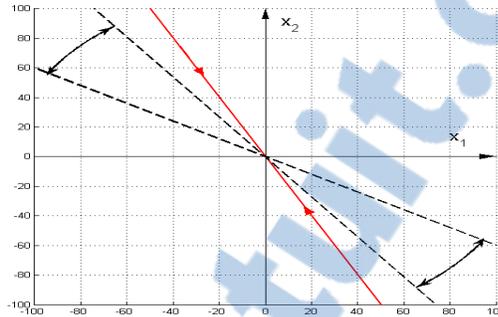


Figure 4.5.4 Limites de placement de la ligne commutative

Alors, il nous reste à vérifier laquelle des lignes commutatives peuvent être utilisées pour que le temps de réponses temporelles soit minimal. Pour cela, on déplace la ligne commutative et donne quelques propositions :

$$x_2(t) = -0,7x_1(t)$$

$$x_2(t) = -x_1(t)$$

$$x_2(t) = -1,3x_1(t)$$

Les trajectoires et les réponses temporelles de ces commandes par modes glissants sont trouvées à l'aide du premier niveau de la plate-forme informatique (voir Figs. 4.5.5 et 4.5.6) :

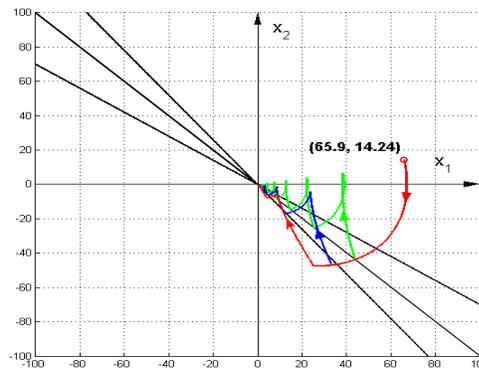


Figure 4.5.5 Démonstration les trajectoires des commandes par modes glissants

Ses réponses temporelles sont montrées dans la figure 4.5.6 :

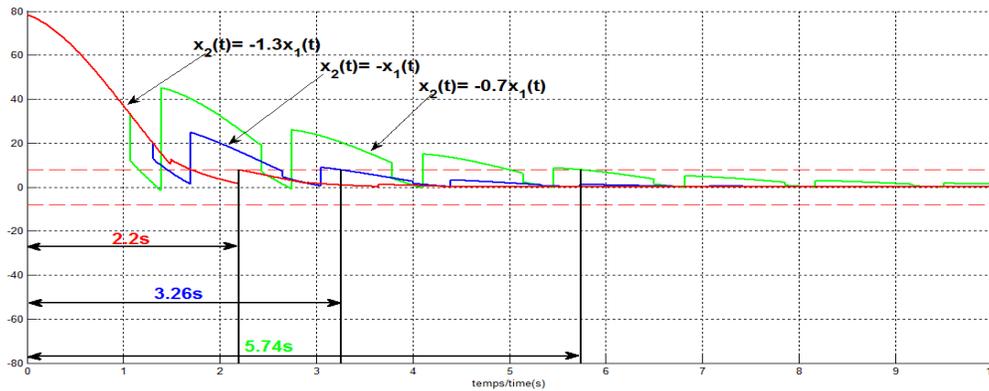


Figure 4.5.6 Réponses temporelles des commandes proposées

Les réponses temporelles du système sont $(2, 2s, 3, 26s, 5, 74s)$. Alors, $x_2(t) = -1,3x_1(t)$ est le meilleur de la proposition ci-dessus. Par conséquent, on peut penser qu'il existe toujours une commande optimale qui respecte le temps de réponse temporelle minimale. C'est une partie intéressante que l'on peut approfondir.

4.6 Conclusion

Lorsqu'on applique la commande par mode glissant sous la base du plan de phase, il existe un temps retard τ causé par la restitution du point initial de stabilité. Par ailleurs, les trajectoires de phase sont trouvées par la méthode d'isoclines qui sont imprédictibles. Dans ce cas, le temps retard τ ne peut pas être éliminé. Pour résoudre ce problème, on propose du remplacement du plan de phase au plan d'état. Des discussions sur le plan de phase et le plan d'état seront présentées au chapitre suivant.

Chapitre 5 Plan d'état et les relations entre plans de phase et d'état

5.1 Introduction

En utilisant, le plan d'état [17, 26, 29, 30, 34, 41, 42, 43, 44, 45, 46], les trajectoires peuvent être trouvées sous forme analytique. Alors, on peut construire une commande en forme explicite, c'est-à-dire que la commande glissante [4, 5, 6, 11, 17, 34, 35, 41, 42, 43, 44, 46].

5.2 Équations d'état en forme d'état

Généralement, pour définir et reconstruire le plan d'état, un système est représenté par une fonction de transfert suivante [17, 29, 30, 41, 46] :

$$G(s) = \frac{Y(s)}{E(s)} = \frac{h_0}{s - s_1} + \dots + \frac{h_n}{s - s_n} \quad (5.1)$$

Son schéma fonctionnel est construit à la manière suivante [12, 17, 29] :

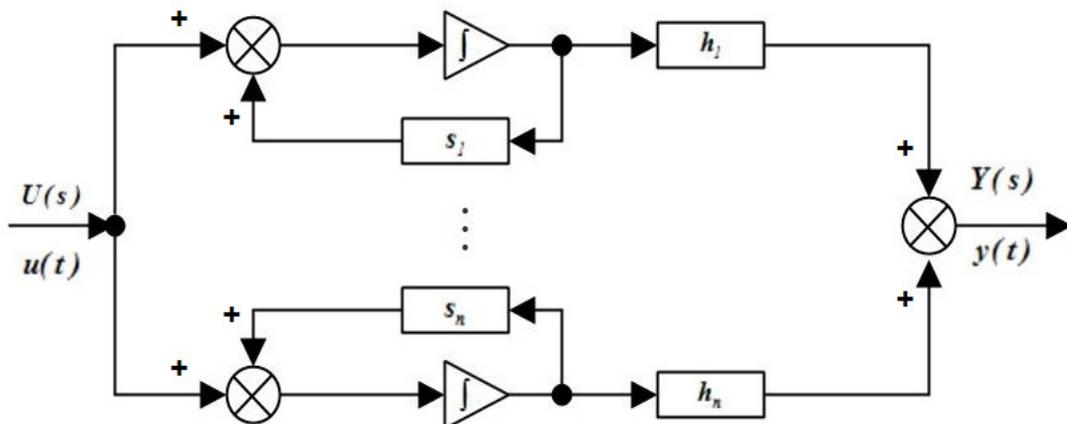


Figure 5.2.1 Schéma fonctionnel d'état

Selon la figure 5.2.1, les variables d'état sont les sorties des intégrateurs, connectés en parallèle, ce qu'expriment les équations d'état [17, 26, 28, 30] :

$$\left\{ \begin{array}{l} \dot{v}_1(t) = \frac{dv_1(t)}{dt} = s_1 v_1(t) + u(t) \\ \dot{v}_2(t) = \frac{dv_2(t)}{dt} = s_2 v_2(t) + u(t) \\ \dots\dots\dots \\ \dot{v}_{n-1}(t) = \frac{dv_{n-1}(t)}{dt} = s_{n-1} v_{n-1}(t) + u(t) \\ \dot{v}_n(t) = \frac{dv_n(t)}{dt} = s_n v_n(t) + u(t) \\ \\ y(t) = h_0 v_1(t) + h_1 v_2(t) + \dots + h_n v_n(t) \end{array} \right. \quad (5.2)$$

Les $v_n(t)$ sont les variables d'état définies dans le plan d'état. Les s_n sont les pôles du système. Les variables d'état sont définies par des pôles du modèle (système) qui diffèrent des variables d'état dans le plan de phase. Par conséquent, il faut faire le passage du plan de phase au plan d'état et ensuite trouver la base de la commande glissante. Grâce aux familles de trajectoires qu'on a utilisées dans la construction de la commande par mode glissant dans le plan de phase, les pôles du modèle (système) peuvent être réels ou complexes conjuguées. Alors, dans la suite de ce chapitre, des discussions seront présentées selon les deux types des pôles du modèle (système).

5.3 Transformation du plan de phase au plan d'état et calcul du plan d'état : pôles réels

Pour faire le passage du plan de phase au plan d'état [12, 17, 29, 32, 34, 46], la méthode de la matrice modale [17, 27, 31, 46] est utilisée. Cette méthode permet le passage des variables d'état $x_n(t)$ vers des variables d'état $v_n(t)$. Alors, la matrice modale M est définie pour faire la transformation.

La démonstration d'application de cette méthode dans notre modèle de second ordre, les équations de transformation sont montrées à la forme suivante :

$$\begin{aligned} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} &= \mathbf{M} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}; \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \\ \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \mathbf{M} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix}; \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} \end{aligned} \quad (5.3)$$

Après les transformations ci-dessus, les variables d'état $x_1(t)$ et $x_2(t)$ dans leurs équations d'état en forme de phase (voir Chap. 3, Éqns. 3.3 et 3.13) peuvent être remplacées par les variables d'état $v_1(t)$ et $v_2(t)$:

$$\begin{aligned} \mathbf{M} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} &= \mathbf{A}\mathbf{M} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{M} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \end{aligned} \quad (5.4)$$

Et ensuite :

$$\begin{aligned} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} &= \mathbf{M}^{-1}\mathbf{A}\mathbf{M} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \mathbf{M}^{-1}\mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{M} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \end{aligned} \quad (5.5)$$

Les nouvelles matrices sont définies :

$$\begin{aligned} \mathbf{A}_p &= \mathbf{M}^{-1}\mathbf{A}\mathbf{M} \\ \mathbf{B}_p &= \mathbf{M}^{-1}\mathbf{B} \\ \mathbf{C}_p &= \mathbf{C}\mathbf{M} \end{aligned} \quad (5.6)$$

Alors, les équations d'état sont trouvées à la forme suivante :

$$\begin{aligned} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} &= \mathbf{A}_p \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \mathbf{B}_p u(t) \\ y(t) &= \mathbf{C}_p \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \end{aligned} \quad (5.7)$$

Avec :

$$\mathbf{A}_p = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}; \mathbf{B}_p = \begin{bmatrix} I \\ I \end{bmatrix}; \mathbf{C}_p = [h_0 \quad h_1] \quad (5.8)$$

Il est possible d'observer que les équations d'état en forme de phase deviennent les équations d'état en forme d'état après cette transformation. Les éléments dans la matrice \mathbf{A}_p sont les pôles du modèle (système). Ils peuvent être calculés par les équations suivantes :

$$\begin{aligned}
s_1 &= \frac{-a_1 + \sqrt{a_1^2 - 4a_0}}{2} \\
s_2 &= \frac{-a_1 - \sqrt{a_1^2 - 4a_0}}{2}
\end{aligned} \tag{5.9}$$

où les paramètres (a_0, a_1) sont les coefficients dans la fonction de transfert du modèle (voir Chap. 3, Éqn. 3.1). Il est facile de trouver que les valeurs propres de la matrice \mathbf{A}_p ont les mêmes valeurs que les valeurs propres de la matrice \mathbf{A} .

Alors, maintenant, il faut trouver la matrice modale \mathbf{M} et sa matrice inversée \mathbf{M}^{-1} pour faire la diagonalisation de la matrice \mathbf{A} et faire le passage vers des variables d'état découplés. Cette matrice modale \mathbf{M} est définie par les vecteurs propres de la matrice \mathbf{A} . Donc, on définit les $\vec{\lambda}_1$ et $\vec{\lambda}_2$ comme des vecteurs propres de la matrice \mathbf{A} . Ils sont définis de la façon suivante :

$$\begin{aligned}
\vec{\lambda}_1 &= \begin{bmatrix} p_{11} \\ p_{21} \end{bmatrix} = \begin{bmatrix} p_{11} \\ \lambda_1 p_{11} \end{bmatrix} = \begin{bmatrix} p_{11} \\ s_1 p_{11} \end{bmatrix} \\
\vec{\lambda}_2 &= \begin{bmatrix} p_{12} \\ p_{22} \end{bmatrix} = \begin{bmatrix} p_{12} \\ \lambda_2 p_{12} \end{bmatrix} = \begin{bmatrix} p_{12} \\ s_2 p_{12} \end{bmatrix}
\end{aligned} \tag{5.10}$$

Ensuite, la matrice modale \mathbf{M} peut être définie à la forme suivante :

$$\mathbf{M} = \begin{bmatrix} \vec{\lambda}_1 & \vec{\lambda}_2 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ \lambda_1 p_{11} & \lambda_2 p_{12} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ s_1 p_{11} & s_2 p_{12} \end{bmatrix} \tag{5.11}$$

et sa matrice inversée \mathbf{M}^{-1} :

$$\mathbf{M}^{-1} = \frac{1}{\det \mathbf{M}} \begin{bmatrix} s_2 p_{12} & -p_{12} \\ -s_1 p_{11} & p_{11} \end{bmatrix} \tag{5.12}$$

En fait, les vecteurs propres de la matrice \mathbf{A} sont infinis. Cependant, on voudrait que la matrice \mathbf{B}_p ait à la forme suivante :

$$\mathbf{B}_p = \mathbf{M}^{-1} \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{5.13}$$

Dans ce cas, les vecteurs propres sont définis et peuvent être calculés par :

$$\frac{1}{\det \mathbf{M}} \begin{bmatrix} s_2 p_{12} & -p_{12} \\ -s_1 p_{11} & p_{11} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{5.14}$$

et puis,

$$\begin{aligned} -p_{12} &= \det \mathbf{M} = (s_2 - s_1)p_{11}p_{12} \\ p_{11} &= \det \mathbf{M} = (s_2 - s_1)p_{11}p_{12} \end{aligned} \quad (5.15)$$

Par conséquent, les éléments dans les vecteurs propres peuvent être trouvés :

$$\begin{aligned} p_{11} &= \frac{I}{s_1 - s_2} \\ p_{12} &= \frac{I}{s_2 - s_1} \end{aligned} \quad (5.16)$$

Remplaçons p_{11} et p_{12} dans les matrices \mathbf{M} et \mathbf{M}^{-1} , on trouve :

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} \vec{\lambda}_1 & \vec{\lambda}_2 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ s_1 p_{11} & s_2 p_{12} \end{bmatrix} = \begin{bmatrix} \frac{I}{s_1 - s_2} & \frac{I}{s_2 - s_1} \\ \frac{s_1}{s_1 - s_2} & \frac{s_2}{s_2 - s_1} \end{bmatrix} \\ \mathbf{M}^{-1} &= \frac{I}{s_1 - s_2} \begin{bmatrix} \frac{s_2}{s_2 - s_1} & \frac{-I}{s_2 - s_1} \\ \frac{-s_1}{s_1 - s_2} & \frac{I}{s_1 - s_2} \end{bmatrix} = \begin{bmatrix} -s_2 & I \\ -s_1 & I \end{bmatrix} \end{aligned} \quad (5.17)$$

Pour cela, les paramètres h_0 et h_1 peuvent être trouvés par l'équation 5.6 :

$$\begin{aligned} h_0 &= \frac{b_0 + b_1 s_1}{s_1 - s_2} \\ h_1 &= \frac{b_0 + b_1 s_2}{s_2 - s_1} \end{aligned} \quad (5.18)$$

où les paramètres (b_0, b_1) sont les coefficients dans le modèle (système) (voir Chap. 3, Éqn. 3.1). Il faut noter que la condition qui est nécessaire pour faire l'application de la méthode de la matrice modale est que $\det(\mathbf{M}) \neq 0$. Parce qu'il faut calculer la matrice \mathbf{M}^{-1} lorsqu'on utilise cette méthode.

Après avoir fait le passage du plan de phase au plan d'état, les conditions initiales peuvent être transformées à ce dernier. Maintenant, il faut trouver la forme analytique générale [17, 32, 46] pour tracer les trajectoires dans le plan d'état. Lorsque l'entrée $u(t)$ est nulle, on exprime l'équation d'état à la forme suivante :

$$\int \frac{dv_{n-1}(t)}{s_{n-1}v_{n-1}(t)} = \int \frac{dv_n(t)}{s_n v_n(t)} \quad (5.19)$$

Par la définition des conditions initiales $v_{n-1}(0)$ et $v_n(0)$, l'équation peut être résolue :

$$\frac{\log[v_{n-1}(t)] - \log[v_{n-1}(0)]}{s_{n-1}} = \frac{\log[v_n(t)] - \log[v_n(0)]}{s_n} \quad (5.20)$$

et puis,

$$\log[v_n(t)] = \frac{s_n \{ \log[v_{n-1}(t)] - \log[v_{n-1}(0)] \}}{s_{n-1}} + \log[v_n(0)] \quad (5.21)$$

Donc, la forme analytique générale est trouvée à la façon suivante :

$$v_n(t) = \left[\frac{v_{n-1}(t)}{v_{n-1}(0)} \right]^{\frac{s_n}{s_{n-1}}} v_n(0) \quad (5.22)$$

Pour notre modèle (système) de second ordre, la forme est :

$$v_2(t) = \left[\frac{v_1(t)}{v_1(0)} \right]^{\frac{s_2}{s_1}} v_2(0) \quad (5.23)$$

où $v_1(0), v_2(0)$ sont les conditions initiales dans le plan d'état. Finalement, on trouve :

$$\frac{dv_2(t)}{dv_1(t)} = f[v_1(t), v_1(0), v_2(0), t] \quad (5.24)$$

Par conséquent, les trajectoires dans le plan d'état peuvent être prédites à chaque instant.

Voici un exemple de la transformation des conditions initiales du plan de phase au plan d'état et la famille de trajectoires trouvée par la forme analytique. Des paramètres du correcteur du modèle sont :

$$K_p = -0,73; K_i = -10,81$$

Pour cela, les coefficients (a_0, a_1, b_0, b_1) sont (voir Chap. 3, Éqns 3.0 et 3.1) :

$$\begin{aligned} a_0 &= -4 \\ a_1 &= 0 \\ b_0 &= -0,27 \\ b_1 &= -4 \end{aligned}$$

Les pôles du modèle (système) peuvent être calculés :

$$s_1 = \frac{-a_1 + \sqrt{a_1^2 - 4a_0}}{2} = 2$$

$$s_2 = \frac{-a_1 - \sqrt{a_1^2 - 4a_0}}{2} = -2$$

ainsi, les valeurs des paramètres de h_0 et h_1 sont trouvées :

$$h_0 = \frac{b_0 + b_1 s_1}{s_1 - s_2} = -2,07$$

$$h_1 = \frac{b_0 + b_1 s_2}{s_2 - s_1} = -1,93$$

Donc, la fonction de transfert et des équations d'état peuvent être trouvées sous les formes suivantes :

$$G_{SL}(s) = \frac{-2,07}{s-2} + \frac{-1,93}{s+2}$$

$$\begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} -2,07 & -1,93 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}$$

Les matrices \mathbf{M} et \mathbf{M}^{-1} sont calculées en vue de faire la transformation :

$$\mathbf{M} = \begin{bmatrix} \frac{1}{s_1 - s_2} & \frac{1}{s_2 - s_1} \\ \frac{s_1}{s_1 - s_2} & \frac{s_2}{s_2 - s_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{-4} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$\mathbf{M}^{-1} = \begin{bmatrix} -s_2 & 1 \\ -s_1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ -2 & 1 \end{bmatrix}$$

Des conditions initiales dans le plan de phase qu'on a choisi ont les valeurs suivantes :

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 10 \\ -21 \end{bmatrix}, \begin{bmatrix} 10 \\ -19 \end{bmatrix}, \begin{bmatrix} -10 \\ 21 \end{bmatrix}, \begin{bmatrix} -10 \\ 19 \end{bmatrix}$$

Les conditions initiales de phase ci-dessus sont transférées au plan d'état par la méthode de la matrice modale :

$$\begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \left(\begin{bmatrix} -1 \\ -41 \end{bmatrix}, \begin{bmatrix} 1 \\ -39 \end{bmatrix}, \begin{bmatrix} 1 \\ 41 \end{bmatrix}, \begin{bmatrix} -1 \\ 39 \end{bmatrix} \right)$$

Alors, avec l'aide de la forme analytique (voir Éqn. 5.23), la famille de trajectoires peut être trouvée dans le plan d'état (voir Fig. 5.3.1 b)).

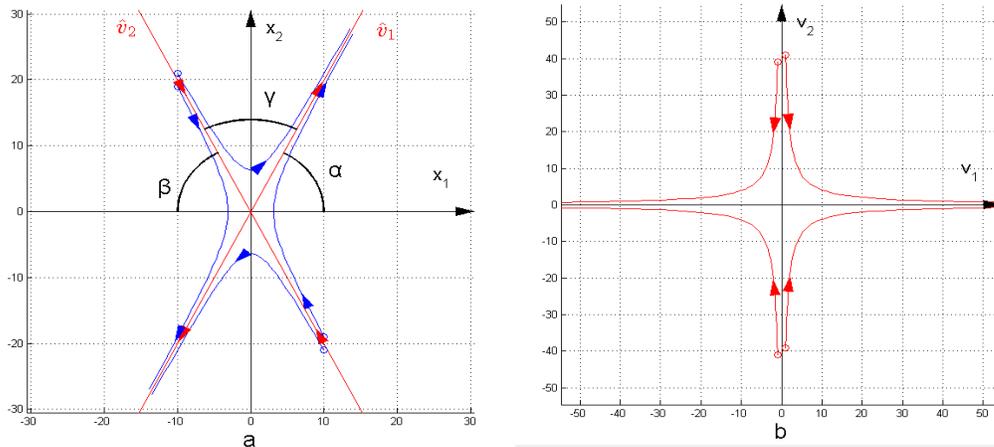


Figure 5.3.1 Famille de trajectoires en forme de col dans deux plans, a) : plan de phase, b) : plan d'état

Nous avons supposé dans le chapitre 3 que les asymptotes de phase sont les projections des coordonnées rectangulaires $[v_1(t), v_2(t)]$ du plan d'état sur le plan de phase. Elles construiraient les coordonnées oblique $[\hat{v}_1(t), \hat{v}_2(t)]$ lorsque l'angle γ est entre $(0 < \gamma < \frac{\pi}{2})$ et $(\frac{\pi}{2} < \gamma < \pi)$. Par ailleurs, lorsque l'angle $\gamma = \frac{\pi}{2}$ est hors de l'intervalle nommé ci-dessus, les asymptotes de phase deviendront les coordonnées rectangulaires $[v_1(t), v_2(t)]$ et resteront sur le plan de phase, ce qui est une propriété plutôt spéciale. Il faut néanmoins signaler que l'utilisation du plan d'état pour représenter la dynamique du système, est nécessaire pour pivoter ces coordonnées à l'angle α pour avoir le point initial du système $[(v_1(0), v_2(0)) \leftarrow (x_1(0), x_2(0))]$.

5.3.1 Transformation géométrique du plan de phase au plan d'état

Supposons maintenant que l'angle $\gamma = \frac{\pi}{2}$. Dans ce cas, le plan de phase et le plan d'état sont sur la même surface (voir Fig. 5.3.2 a), c) et e)). Alors, on peut faire une rotation d'angle α pour trouver le point initial du système dans le plan d'état (voir Fig. 5.3.2) :

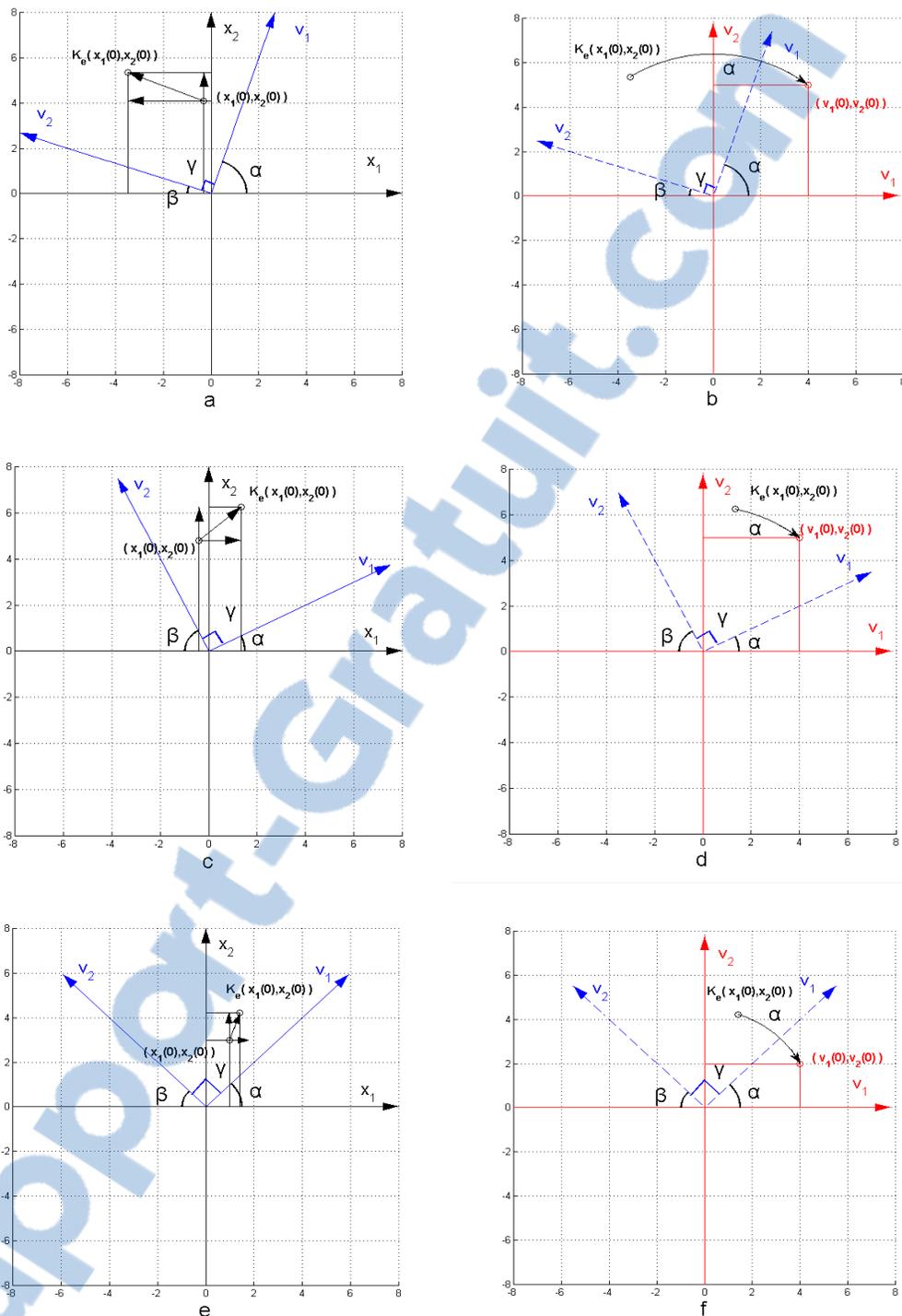


Figure 5.3.2 Élongation et rotation du point initial du système, a) : Élongation des coordonnées du point initial ($0 < \beta < \alpha < \frac{\pi}{2}$), b) : Rotation à sens horaire d'un angle α ($0 < \beta < \alpha < \frac{\pi}{2}$), c) :

Élongation des coordonnées du point initial ($0 < \alpha < \beta < \frac{\pi}{2}$), d) : Rotation à sens horaire d'un

angle α ($0 < \alpha < \beta < \frac{\pi}{2}$), e) : Élongation des coordonnées du point initial ($\alpha = \beta = \frac{\pi}{4}$), et f) :

Rotation à sens horaire d'un angle α ($\alpha = \beta = \frac{\pi}{4}$)

Par la figure 5.3.2, on montre alors qu'on fait une élongation des coordonnées du point initial et ensuite, une rotation à sens horaire d'angle α pour redéfinir le point initial. Après cette rotation, les asymptotes de phase deviendront les coordonnées rectangulaires $[v_1(t), v_2(t)]$ et on place la représentation dynamique du point initial sur le plan d'état (voir Éqn. 5.25) :

$$\begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix} = K_e(x_1(0), x_2(0)) R_r(\alpha) \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (5.25)$$

où $K_e(x_1(0), x_2(0))$ représente l'élongation des coordonnées du point initial de phase, $R_r(\alpha)$ représente la rotation à sens horaire d'angle α .

Faisons maintenant un examen plus précis des différentes hypothèses faites sur les angles γ qu'elles sont comprises dans $\left(0 < \gamma < \frac{\pi}{2}\right)$ ou $\left(\frac{\pi}{2} < \gamma < \pi\right)$. Dans ce cas, plusieurs étapes de transformation : perspective, projection et élongation du point initial devraient être faites sur les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ et les coordonnées du point initial.

Les étapes essentielles sont les suivantes :

- 1) La démonstration de la perspective,
- 2) La démonstration de la rotation,
- 3) La démonstration de la projection : pour $\left(0 < \gamma < \frac{\pi}{2}\right)$, ou $\left(\frac{\pi}{2} < \gamma < \pi\right)$,
- 4) La démonstration d'élongation des coordonnées du point initial,
- 5) La démonstration du théorème de la transformation complexe.

Supposons maintenant $\gamma \in \left(0 < \gamma < \frac{\pi}{2}\right)$, la première étape est évidente, alors on fait une perspective (voir Fig. 5.3.3) :

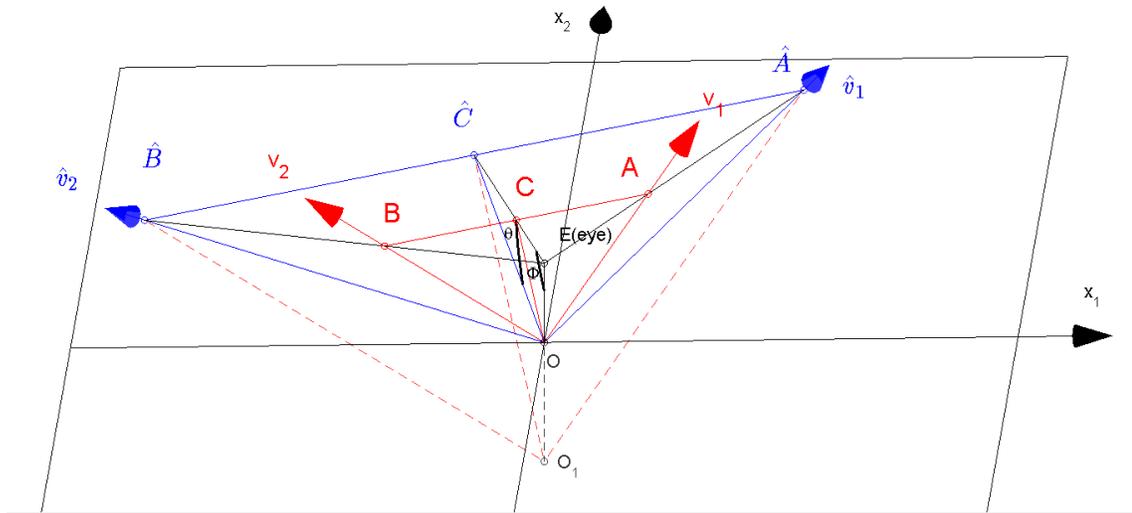


Figure 5.3.3 Démonstration de la perspective et de l'angle de la rotation θ

Dans la figure 5.3.3, le point de la perspective est le point E . On définit la ligne OE , qui est perpendiculaire au plan d'état et ensuite, on procède une rotation du plan d'oblique $[\hat{v}_1(t), \hat{v}_2(t)]$ pour qu'elle devienne le plan d'état. Cet angle de la rotation est θ .

Démonstration la perspective (voir Fig. 5.3.4) dans le plan de phase et le plan d'état :

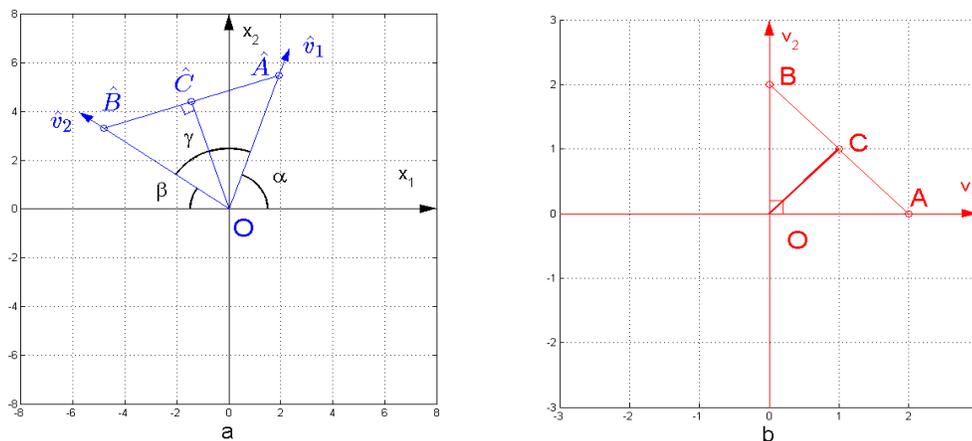


Figure 5.3.4 Démonstration la bissectrice OC et la bissectrice OC , a) : Bissectrice des asymptotes de phase OC , b) : Bissectrice OC des coordonnées d'état $[v_1(t), v_2(t)]$

Maintenant, voyons la figure 5.3.4 b). Dans le plan d'état, on définit deux lignes OA et OB dans les coordonnées d'état $[v_1(t), v_2(t)]$ respectivement où $OA=OB$. Le point C est le point d'intersection de la bissectrice OC et la ligne AB . Alors, il est facile de trouver que cette bissectrice est perpendiculaire à la ligne AB avec $OC = \frac{1}{2} AB$. Lorsqu'on fait la perspective du plan d'état (voir Fig. 5.3.3), les lignes OA et OB sont projetées dans le plan de phase sur les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ où les points \hat{A}, \hat{B} sont les projections des points A, B respectivement. La bissectrice OC est projetée dans le plan de phase pour construire la ligne $O\hat{C}$. Dans ce cas, la bissectrice $O\hat{C}$ est perpendiculaire à la ligne $\hat{A}\hat{B}$. Alors, on peut écrire que :

$$\cot\left(\frac{\gamma}{2}\right) = \frac{O\hat{C}}{\frac{1}{2}\hat{A}\hat{B}} \Rightarrow O\hat{C} = \frac{1}{2}\hat{A}\hat{B} \cot\left(\frac{\gamma}{2}\right) \quad (5.26)$$

Maintenant, avec l'angle de la perspective présentée à la figure 5.3.3, on peut trouver une relation géométrique entre le plan de phase et le plan d'état :

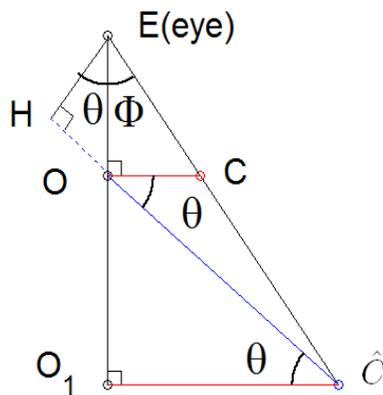


Figure 5.3.5 Définition de l'angle θ par la relation géométrique

Dans la figure 5.3.5, le point E est le point de la perspective, le point O est le point d'origine du plan de phase et du plan d'état. L'angle de la rotation est θ et puis l'angle de la perspective est Φ . La ligne OC est la bissectrice des coordonnées d'état $[v_1(t), v_2(t)]$ dans le plan d'état. Sa projection dans le plan de phase est la ligne $O\hat{C}$ qui est la bissectrice des asymptotes de phase aussi.

Dans ce cas, on peut définir l'angle de la rotation θ entre la bissectrice OC et la bissectrice $O\hat{C}$. Et ensuite, trouver cet angle entre la bissectrice $O\hat{C}$ et la ligne $O_1\hat{C}$. Finalement, on peut trouver :

$$\begin{aligned}\cos(\theta) &= \frac{O_1\hat{C}}{O\hat{C}} \\ \theta &= \arccos\left(\frac{O_1\hat{C}}{O\hat{C}}\right)\end{aligned}\quad (5.27)$$

Le plan où on trouve la ligne $O_1\hat{C}$ est parallèle du plan d'état. Alors, on peut trouver les triangles similaires $\Delta OAB \square \Delta O_1\hat{A}\hat{B}$ (voir Fig. 5.3.3). Dans ce cas, la ligne $O_1\hat{C}$ peut être trouvée par la théorie de la perspective géométrique :

$$\begin{aligned}\frac{AB}{\hat{A}\hat{B}} &= \frac{OC}{O_1\hat{C}} \\ O_1\hat{C} &= \frac{OC * \hat{A}\hat{B}}{AB} = \frac{1}{2} \hat{A}\hat{B}\end{aligned}\quad (5.28)$$

Finalement l'angle de rotation des coordonnées obliques est suivant :

$$\theta = \arccos\left(\frac{O_1\hat{C}}{O\hat{C}}\right) = \arccos\left(\frac{\frac{1}{2}\hat{A}\hat{B}}{\frac{1}{2}\hat{A}\hat{B}\cot\left(\frac{\gamma}{2}\right)}\right) = \arccos\left(\frac{1}{\cot\left(\frac{\gamma}{2}\right)}\right) = \arccos\left(\tan\left(\frac{\gamma}{2}\right)\right)\quad (5.29)$$

En ce qui concerne de l'angle de la perspective Φ , on peut le trouver par l'utilisation de la loi des cosinus (voir Fig.5.3.5) :

$$\begin{aligned}O\hat{C}^2 &= EO^2 + E\hat{C}^2 - 2EOE\hat{C} \cos(\Phi) \\ &\Downarrow \\ \Phi &= \arccos\left(\frac{EO^2 + E\hat{C}^2 - O\hat{C}^2}{2EOE\hat{C}}\right)\end{aligned}\quad (5.30)$$

où : $E\hat{C}$ est la distance entre le point de la perspective E et la projection \hat{C} , EO est la distance entre le point de la perspective E et le point d'original, et $O\hat{C}$ est la distance entre de la projection \hat{C} et le point d'original. La distance entre le point de la perspective E et le plan de phase est EH .

Pour la situation $\gamma \in \left(\frac{\pi}{2} < \gamma < \pi\right)$, on peut trouver une relation géométrique similaire, voir ci-dessus, à la figure 5.3.5 :

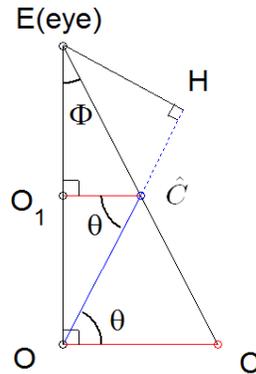


Figure 5.3.6 Définition de l'angle θ par la relation géométrique lorsque $\gamma \in \left(\frac{\pi}{2} < \gamma < \pi\right)$

Dans la figure 5.3.6, l'angle de la rotation est aussi défini comme θ . Il peut être trouvé entre la bissectrice des asymptotes de phase $O\hat{C}$ et la bissectrice des coordonnées d'état OC . De même, on définit que la ligne OE est perpendiculaire au plan d'état avec le point E est le point de la perspective. De plus, le point O_1 se trouve entre les points E et O sur la ligne OE . Dans ce cas, le même angle θ peut être trouvé entre la bissectrice $O\hat{C}$ et la ligne $O_1\hat{C}$. Alors, l'angle θ satisfait la relation dans l'équation 5.27 et il peut être trouvé par l'équation 5.29. Similairement, l'angle de la perspective Φ peut être trouvé par l'équation 5.30 où il est défini par les lignes $E\hat{C}$, EO et $O\hat{C}$. Ces lignes représentent les distances entre le point de la perspective E , le point d'original O et le point de la projection \hat{C} . De plus, EH représente la distance entre le point de la perspective E et le plan de phase.

Lorsque la rotation de plan d'oblique $[\hat{v}_1(t), \hat{v}_2(t)]$ est définie, on peut faire la perspective du plan d'état et trouver la projection du point initial d'état dans le plan de phase :

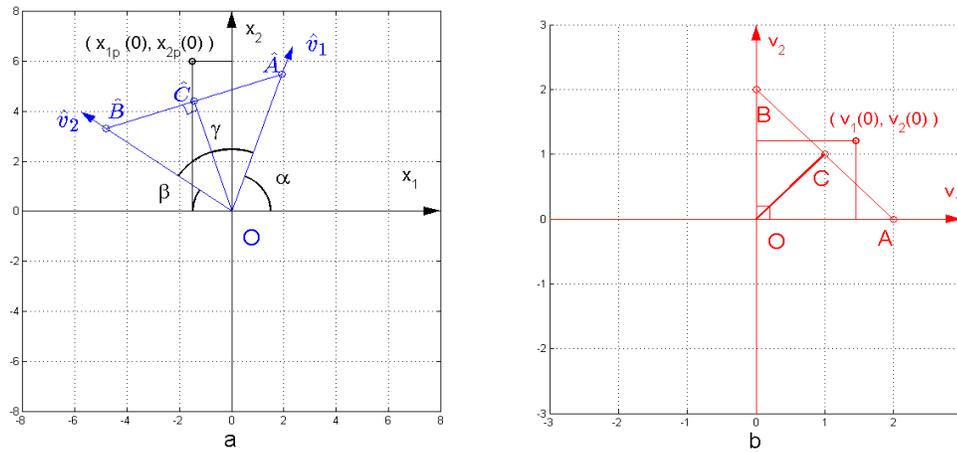


Figure 5.3.7 Démonstration de la projection du point initial dans le plan de phase, a) : Projection du point initial d'état $(v_1(0), v_2(0))$ dans le plan de phase, b) : Point initial $(v_1(0), v_2(0))$ dans le plan d'état

Avec la méthode de la perspective, on peut trouver la projection du point initial d'état $(v_1(0), v_2(0))$ dans le plan de phase comme le point $(x_{1p}(0), x_{2p}(0))$.

S'il est nécessaire, on peut faire une élongation des coordonnées du point initial de phase $(x_1(0), x_2(0))$ pour trouver cette projection (voir Fig. 5.3.8) :

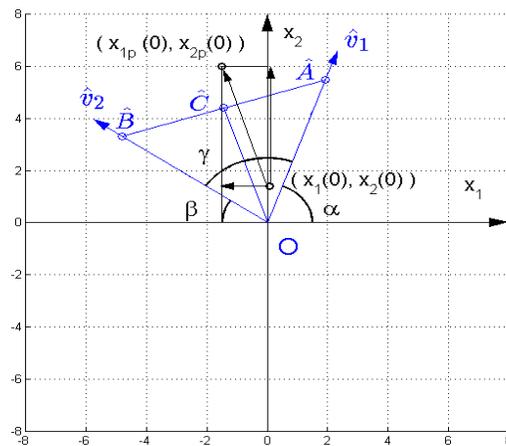


Figure 5.3.8 Démonstration de l'élongation des coordonnées du point initial de phase

Finalement, un théorème de la transformation complexe est trouvé d'où on fait une élongation des coordonnées du point initial $(x_1(0), x_2(0))$ et ensuite, une transformation inverse de la perspective pour redéfinir le point initial. Après cette transformation complexe, les coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ deviendront les coordonnées rectangulaires $[v_1(t), v_2(t)]$ et on place la représentation dynamique du point initial sur le plan d'état (voir Éqn.5.31) :

$$\begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix} = K_e(x_1(0), x_2(0)) T_p^{-1}(\theta) \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (5.31)$$

où $K_e(x_1(0), x_2(0))$ est l'élongation des coordonnées du point initial $(x_1(0), x_2(0))$ et $T_p^{-1}(\theta)$ est la transformation inverse de la perspective avec l'angle de la rotation des coordonnées obliques $[\hat{v}_1(t), \hat{v}_2(t)]$ est θ .

D'après la démonstration de cette transformation complexe, comme ci-dessous, on peut appliquer cette méthode au système ayant le modèle de paramètres :

$$1) : K_p = -3,44, K_i = -5,41$$

Pour cela, les pôles du modèle (système) sont :

$$s_1 = 2, s_2 = -1$$

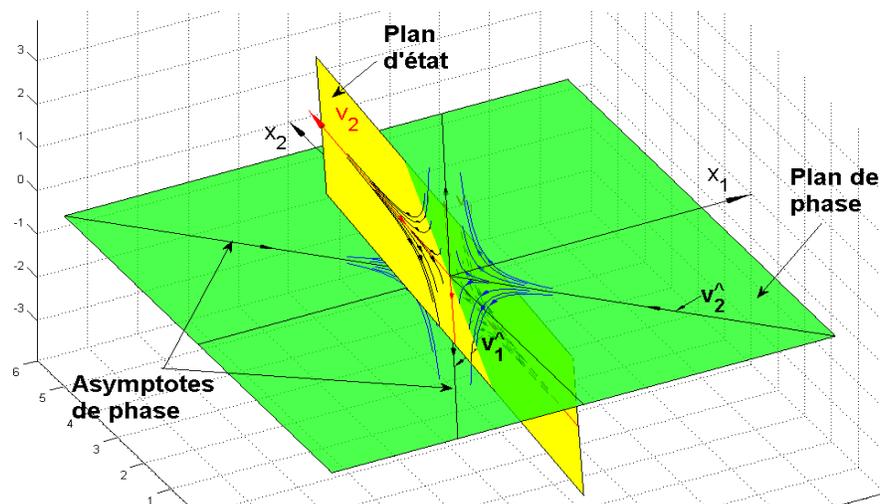


Figure 5.3.9 Démonstration du plan d'état et du plan de phase dans l'espace avec pôles réels :

$$s_1 = 2 \text{ et } s_2 = -1$$

Démonstration d'un autre exemple : Supposons maintenant que les paramètres du modèle (système) sont :

$$2) : K_p = -0,73, K_i = -0,68$$

Donc les pôles du modèle (système) sont :

$$s_1 = 0,5, s_2 = -0,5$$

Démonstration les deux plans dans l'espace :

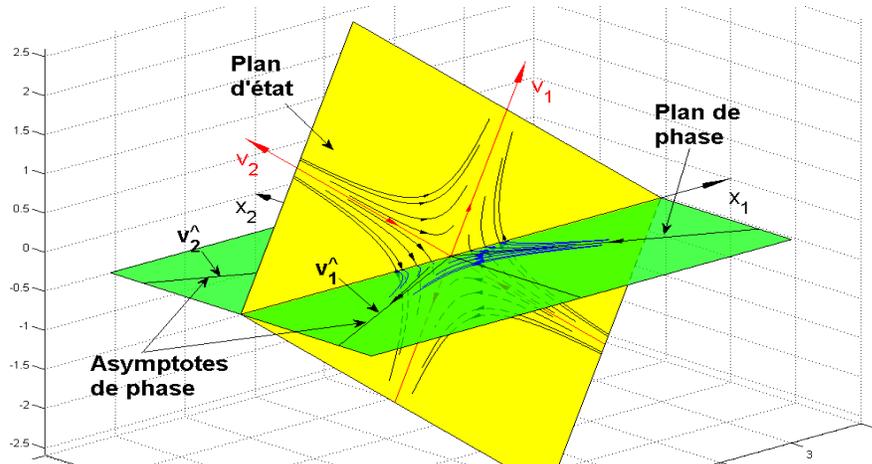


Figure 5.3.10 Démonstration du plan d'état et du plan de phase dans l'espace avec pôles réels :

$$s_1 = 0,5 \text{ et } s_2 = -0,5$$

Le deuxième niveau de la plate-forme informatique est construit d'où la transformation du plan de phase au plan d'état et la visualisation du plan de phase et du plan d'état dans l'espace. Une description détaillée de ce niveau de la plate-forme informatique est présentée à l'**Annexe C : Plate-forme des calculs des transformations géométriques**.

Le plan d'état trouvé dans ce cas a des pôles réels et ses trajectoires seront comme une partie de la base pour construire la commande glissante. L'autre partie de la base est trouvée par les pôles complexes conjugués [17, 29, 35, 47] où les trajectoires sont formées de spirales.

5.4 Transformation du plan de phase au plan d'état et calcul du plan d'état : cas pôles complexes conjugués

Lorsque les pôles du modèle (système) sont des nombres complexes conjugués, ils sont définis à la forme suivante:

$$\begin{aligned} s_1 &= \alpha + j\beta \\ s_2 &= \alpha - j\beta \end{aligned} \quad (5.32)$$

Dans ce cas, les paramètres h_0 et h_1 sont définis comme :

$$h_0 = \gamma - j\delta, h_1 = \gamma + j\delta \quad (5.33)$$

Les paramètres γ et δ peuvent être calculés par les équations ci-dessous :

$$\begin{aligned} \mathbf{C}_p &= \mathbf{C}\mathbf{M} \\ \mathbf{C} &= [b_0 \quad b_1] \end{aligned} \quad (5.34)$$

$$\mathbf{C}_p = [b_0 \quad b_1] \begin{bmatrix} \frac{1}{j2\beta} & \frac{1}{-j2\beta} \\ \frac{\alpha + j\beta}{j2\beta} & \frac{\alpha - j\beta}{-j2\beta} \end{bmatrix} = \begin{bmatrix} \frac{b_1\beta - j(b_0 + b_1\alpha)}{2\beta} & \frac{b_1\beta + j(b_0 + b_1\alpha)}{2\beta} \end{bmatrix} \quad (5.35)$$

Ensuite, ils sont trouvés à la forme suivant (voir Chap. 3, Éqns 3.0 et 3.1):

$$\begin{aligned} \gamma &= \frac{b_1}{2} \\ \delta &= \frac{b_0 + b_1\alpha}{2\beta} \end{aligned} \quad (5.36)$$

On définit les variables d'état comme $\tilde{v}_1(t)$ et $\tilde{v}_2(t)$. Alors, les équations d'état peuvent être écrites :

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{v}}_1(t) \\ \dot{\tilde{v}}_2(t) \end{bmatrix} &= \begin{bmatrix} \alpha + j\beta & 0 \\ 0 & \alpha - j\beta \end{bmatrix} \begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \\ y(t) &= [\gamma - j\delta \quad \gamma + j\delta] \begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} \end{aligned} \quad (5.37)$$

Si $u(t) = 0$,

$$\frac{d\tilde{v}_2(t)}{(\alpha - j\beta)\tilde{v}_2(t)} = \frac{d\tilde{v}_1(t)}{(\alpha + j\beta)\tilde{v}_1(t)} \quad (5.38)$$

La forme analytique devient:

$$\begin{aligned} \tilde{v}_2(t) &= \left(\frac{\tilde{v}_1(t)}{\tilde{v}_1(0)} \right)^{\frac{\alpha - j\beta}{\alpha + j\beta}} \tilde{v}_2(0) \\ \text{ou : } \tilde{v}_2(t) &= \tilde{v}_2(0) \left(\frac{\tilde{v}_1(t)}{\tilde{v}_1(0)} \right)^{\frac{\alpha^2 - \beta^2}{\alpha^2 + \beta^2}} \left(\frac{\tilde{v}_1(t)}{\tilde{v}_1(0)} \right)^{\frac{-2\alpha\beta j}{\alpha^2 + \beta^2}} \end{aligned} \quad (5.39)$$

où : $\tilde{v}_1(0)$ et $\tilde{v}_2(0)$ sont des conditions initiales.

Si on résout cette équation (voir Éqn.5.39), nous pourrions obtenir dans la solution les trajectoires complexes conjuguées. Dans ce cas, si on définit la partie réelle et la partie imaginaire, on peut utiliser la même méthode dans l'explication ci-dessous pour trouver les trajectoires complexes conjuguées. Cette méthode est utilisée pour trouver les trajectoires sur la surface aux valeurs réelles d'état $[z_1(t), z_2(t)]$ (pour plus d'information, voir citation [48] : « Stanislaw Tarasiewicz : Rapport théorique sur la transformation complexe »).

Maintenant, on appelle encore la forme suivante :

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{v}}_1(t) \\ \dot{\tilde{v}}_2(t) \end{bmatrix} &= \begin{bmatrix} \alpha + j\beta & 0 \\ 0 & \alpha - j\beta \end{bmatrix} \begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} + \begin{bmatrix} I \\ I \end{bmatrix} u(t) \\ y(t) &= [\gamma - j\delta \quad \gamma + j\delta] \begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} \end{aligned} \quad (5.40)$$

Et pour se ramener à des expressions réelles, on prend comme nouvelles variables :

$$\begin{aligned} z_1(t) &= \tilde{v}_1(t) + \tilde{v}_2(t) \\ z_2(t) &= j[\tilde{v}_1(t) - \tilde{v}_2(t)] \end{aligned} \quad (5.41)$$

Et le modèle devient en forme suivant :

$$\begin{aligned} \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} &= \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u(t) \\ y(t) &= [\gamma \quad -\delta] \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \end{aligned} \quad (5.42)$$

Ce type de changement de variables d'état met en évidence leurs parties réelles et imaginaires, et à l'aide de la méthode des isoclines (voir Chap. 3.3), on peut tracer les trajectoires numériquement.

Avec un point de vue sur la transformation complexe, on peut alors définir les coordonnées polaires [48] sur la surface aux valeurs réelles d'état $[z_1(t), z_2(t)]$. Dans ce cas, d'axes le point initial P_1 est caractérisé par :

$$\begin{cases} z_1(t) = R(t) \cos[\Theta(t)] \\ z_2(t) = R(t) \sin[\Theta(t)] \end{cases} \quad (5.43)$$

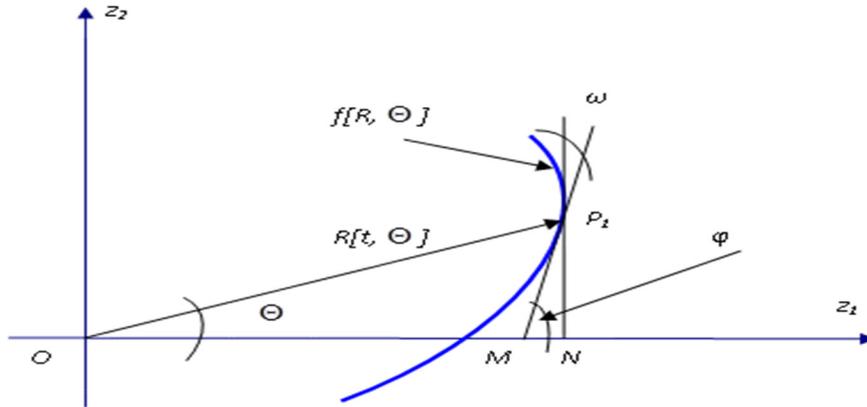


Figure 5.4.1 Une portion de l'arc de spirale du système, où : $R(t)$ est la distance à l'origine, et $\Theta(t)$ est l'angle polaire

À partir de la condition initiale du système, $\{P_1(z_1(t), z_2(t))\}$ on introduit deux triangles (ΔP_1NM et ΔONP_1) respectivement. On sait que :

$$\varphi = \Theta + \omega \rightarrow \omega = \varphi - \Theta \quad (5.44)$$

Et

$$R^2(t) = z_1^2(t) + z_2^2(t) \rightarrow R \frac{dR(t)}{dt} = z_1(t) \frac{dz_1(t)}{dt} + z_2(t) \frac{dz_2(t)}{dt} \quad (5.45)$$

On remarque que la pente de la tangente à la trajectoire $\{f[R(t), \Theta(t)]\}$ passant au point $\{P_1(z_1(t), z_2(t))\}$ est $\arctan(\varphi)$.

D'autre part, on obtient :

$$\begin{cases} \frac{dz_1(t)}{dt} = \alpha \cdot R(t) \cos[\Theta(t)] + \beta \cdot R(t) \sin[\Theta(t)] \\ \frac{dz_2(t)}{dt} = -\beta \cdot R(t) \cos[\Theta(t)] + \alpha \cdot R(t) \sin[\Theta(t)] \end{cases} \quad (5.46)$$

où $s_{1/2} = \alpha \pm j\beta$.

Maintenant, on trouve :

$$\frac{dR(t)}{dt} = R e^{[s_1]}. R(t) \quad (5.47)$$

Dans l'analyse qui suit, on définit l'angle polaire selon les dérivations de :

$$\frac{d \left[\frac{z_2(t)}{z_1(t)} \right]}{dt} = \frac{z_1(t) \frac{dz_2(t)}{dt} - z_2(t) \frac{dz_1(t)}{dt}}{z_1^2(t)} \quad (5.48)$$

Et puis,

$$\frac{d}{dt} [\tan \Theta(t)] = \left\{ 1 + \left[\frac{z_2(t)}{z_1(t)} \right]^2 \right\} \frac{d\Theta(t)}{dt} \Rightarrow \frac{d\Theta(t)}{dt} = \frac{z_1(t) \frac{dz_2(t)}{dt} - z_2(t) \frac{dz_1(t)}{dt}}{z_1^2(t) + z_2^2(t)} \quad (5.49)$$

En conclusion, chaque point de la surface réelle $[z_1(t), z_2(t)]$ (voir Fig. 5.4.1) est déterminé par les équations ci-dessous. Lorsque le temps t croît, le point $\{P_1(z_1(t), z_2(t))\}$ décrit une trajectoire dans une surface aux valeurs réelles du plan d'état à la forme suivante :

$$R[t, \Theta(t)] = \frac{\exp\left(\frac{\beta}{\alpha}\right)}{R(0)} \exp[\Theta(t) + \Theta(0)] \quad (5.50)$$

où : $R(0) = \sqrt{z_1^2(0) + z_2^2(0)}$ et $\Theta(0) = \arctan\left(\frac{z_2(0)}{z_1(0)}\right)$.

Par suite, avec les conditions initiales admissibles $(z_1(0), z_2(0))$, on construit, sur une surface aux valeurs réelles, une famille des trajectoires du système. Il est évident qu'à partir de tous les pôles complexes du système commandé, on obtient la famille des surfaces multiples d'état.

Voici la démonstration d'une famille de trajectoires en forme de foyer dans le plan de phase, la famille de trajectoires qui sont tracées par la méthode des coordonnées polaires sur la surface aux valeurs réelles $[z_1(t), z_2(t)]$ et les trajectoires complexes conjuguées présentées par la partie réelle et la partie imaginaire (voir Fig. 5.4.2). Les conditions initiales de phase sont transformées aux variables d'état $\tilde{v}_1(t), \tilde{v}_2(t)$ et transformées aux variables d'état $z_1(t), z_2(t)$:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ -5 \end{bmatrix}, \begin{bmatrix} -5 \\ 5 \end{bmatrix}, \begin{bmatrix} -5 \\ -5 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 6,6 + 5,02j \\ 6,6 - 5,02j \end{bmatrix}, \begin{bmatrix} -3,4 + 5,02j \\ -3,4 - 5,02j \end{bmatrix}, \begin{bmatrix} 3,4 - 5,02j \\ 3,4 + 5,02j \end{bmatrix}, \begin{bmatrix} -6,6 - 5,02j \\ -6,6 + 5,02j \end{bmatrix}$$

$$\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} 13,2 \\ -10,04 \end{bmatrix}, \begin{bmatrix} -6,8 \\ -10,04 \end{bmatrix}, \begin{bmatrix} 6,8 \\ 10,04 \end{bmatrix}, \begin{bmatrix} -13,2 \\ 10,04 \end{bmatrix}$$

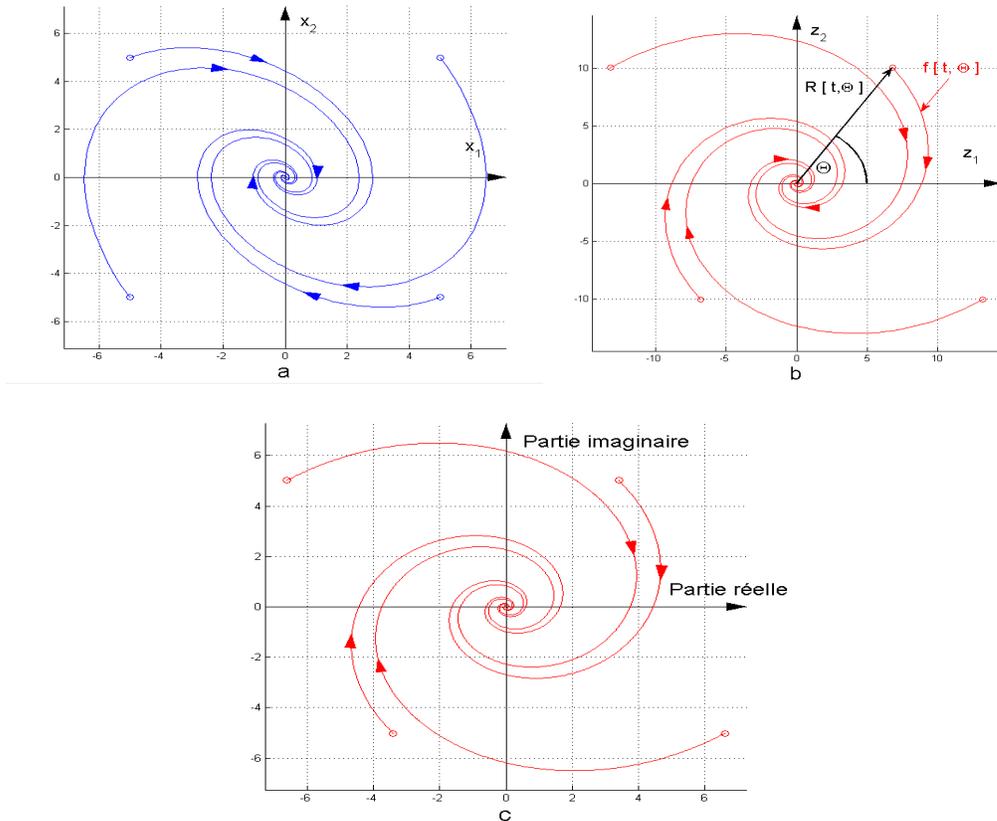


Figure 5.4.2 Famille de trajectoires en forme de foyer (spirales), a) : Trajectoires dans le plan de phase, b) : Trajectoires réelles dans le plan $[z_1(t), z_2(t)]$, c) : Trajectoires complexes conjuguées

$(\tilde{v}_1(t), \tilde{v}_2(t))$ présentées par la partie réelle $\text{Re}[f(R(t), \Theta(t))]$ et la partie imaginaire

$$\text{Im}[f(R(t), \Theta(t))]$$

Dans la figure 5.4.2, les trajectoires de phase sont présentées dans la figure 5.4.2 a). Et dans la figure 5.4.2 b), les trajectoires dans le plan $[z_1(t), z_2(t)]$ sont trouvées sous forme analytique dans l'équation 5.50 par la méthode des coordonnées polaires où les trajectoires sont définies par les paramètres du rayon $R[t, \Theta(t)]$ et de l'angle $\Theta(t)$. Alors, les trajectoires de cette surface aux valeurs réelles sont analytiques et réelles. Il est possible qu'on utilise ces trajectoires réelles pour faire la construction de la commande glissante. À la figure 5.4.2 c), les trajectoires complexes conjuguées sont représentées par la partie réelle et la partie imaginaire. Elles sont la moitié de la forme des trajectoires réelles, mais tracées dans le plan complexe. Il faut utiliser la sphère de **Riemann** [46, 47, 49, 50] pour trouver ces trajectoires dans le domaine réel.

5.4.1 Transformation géométrique des trajectoires complexes conjuguées à la sphère de Riemann

On définit la sphère de **Riemann** à la forme suivante :

$$[v_{R1}(t), v_{R2}(t), v_{R3}(t)] \in \mathfrak{R} : v_{R1}^2(t) + v_{R2}^2(t) + [v_{R3}(t) - R]^2 = R^2 ; R \in \mathfrak{R} \quad (5.51)$$

où $v_{R1}(t), v_{R2}(t), v_{R3}(t)$ sont les coordonnées de la sphère dans l'espace, R est le rayon. On définit une application de la sphère à la façon suivante : à tout point S sur la surface de la sphère, différent du pôle nord N et on associe ce point avec l'unique point C dans le plan d'état complexe par la ligne NC (voir Fig. 5.4.3) :

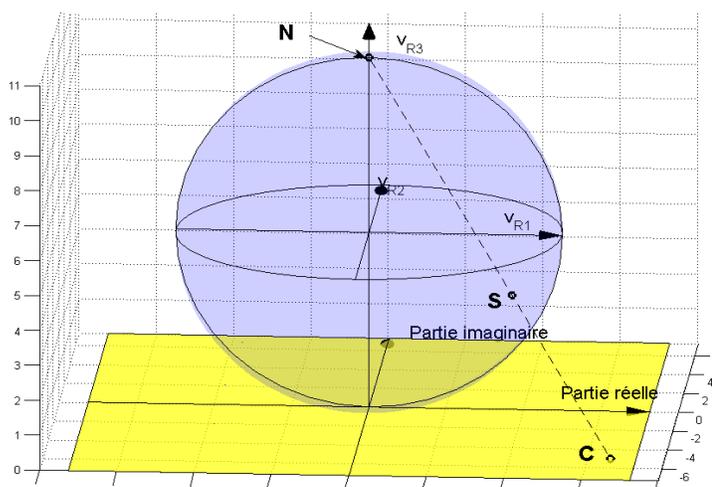


Figure 5.4.3 Définition la sphère de **Riemann**

Cette application est la projection stéréographique où chaque point qui se trouve sur cette surface représente d'un point unique dans le plan d'état complexe. Le pôle Nord de la sphère présente le point infini. Pour cela, on peut trouver les projections des trajectoires complexes conjuguées à la sphère de **Riemann**, et on trouve les trajectoires réelles (pour plus d'information, voir **Annexe C : Plate-forme des calculs des transformations géométriques**). L'équation pour faire la projection stéréographique peut être trouvée de la manière suivante :

$$\frac{v_{R1}(t)}{P_{Reel}(t)} = \frac{v_{R2}(t)}{P_{Imag}(t)} = \frac{v_{R3}(t) - 2R}{-2R} \quad (5.52)$$

où $P_{Reel}(t)$ représente la partie réelle des trajectoires complexes conjuguées, $P_{Imag}(t)$ représente la partie imaginaire. Alors on peut trouver :

$$\begin{aligned} v_{R2}(t) &= \frac{P_{Imag}(t)}{P_{Reel}(t)} v_{R1}(t) \\ v_{R3}(t) - R &= R - \frac{2Rv_{R1}(t)}{P_{Reel}(t)} \end{aligned} \quad (5.53)$$

Remplaçons les coordonnées $v_{R2}(t)$ et $v_{R3}(t)$ dans l'équation de **Riemann** (voir Éqn. 5.51) par la forme ci-dessus, on trouve :

$$v_{R1}^2(t) + \left[\frac{P_{Imag}(t)}{P_{Reel}(t)} v_{R1}(t) \right]^2 + \left[R - \frac{2Rv_{R1}(t)}{P_{Reel}(t)} \right]^2 = R^2 \quad (5.54)$$

Finalement, l'équation de la projection stéréographique est trouvée :

$$\begin{aligned} v_{R1}(t) &= \frac{4R^2 P_{Reel}(t)}{P_{Reel}^2(t) + P_{Imag}^2(t) + 4R^2} \\ v_{R2}(t) &= \frac{4R^2 P_{Imag}(t)}{P_{Reel}^2(t) + P_{Imag}^2(t) + 4R^2} \\ v_{R3}(t) &= \frac{2R [P_{Reel}^2(t) + P_{Imag}^2(t)]}{P_{Reel}^2(t) + P_{Imag}^2(t) + 4R^2} \end{aligned} \quad (5.55)$$

Où $P_{Reel}(t)$ et $P_{Imag}(t)$ sont les parties réelle et imaginaire des trajectoires complexes conjuguées respectivement. R est le rayon de la sphère de **Riemann**.

Voici un exemple de la transformation géométrique des trajectoires complexes conjuguées (voir Fig. 5.4.2 c)) à la sphère de **Riemann**. Le rayon de la sphère est défini $R = 5$:

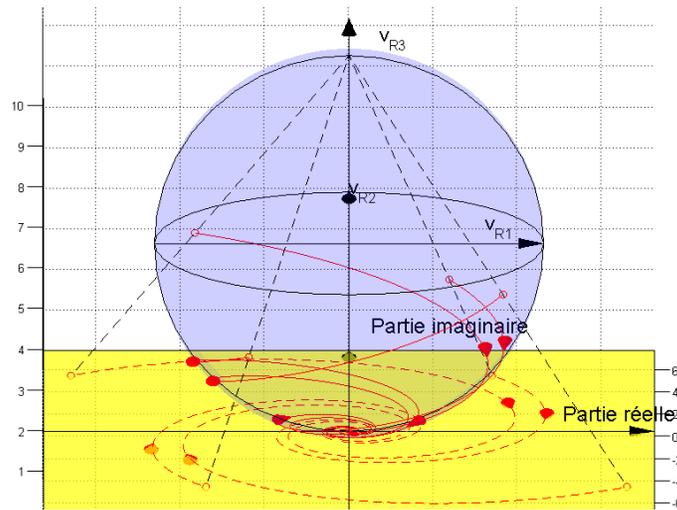


Figure 5.4.4 Démonstration la transformation géométrique à la sphère de **Riemann**.

Les trajectoires pointillées rouges sont les trajectoires complexes conjuguées présentées par la partie réelle et la partie imaginaire dans le plan d'état complexe. Les trajectoires réelles sont les trajectoires continues rouges trouvées dans la sphère. Par conséquent, la sphère de **Riemann** devient d'une partie de la base pour construire la commande glissante. Ces trajectoires réelles peuvent être utilisées pour construire cette commande.

5.5 Conclusion

On remplace le plan de phase par le plan d'état où les trajectoires sont en formes analytiques, réelles ou complexes conjuguées. Par conséquent, on peut avoir la connaissance des trajectoires pour construire la commande glissante. Il faut que toutes les trajectoires soient réelles pour faire la construction. Alors, on fait les transformations géométriques pour trouver la base de la commande glissante. Le plan d'état trouvé avec des pôles réels et la sphère de **Riemann** compose de cette base. La démonstration de la commande glissante est présentée au chapitre suivant.

Chapitre 6 Commande glissante dans le plan d'état

6.1 Introduction

On a construit le troisième niveau de la plate-forme informatique pour faire la démonstration et la visualisation d'application de la commande glissante. Une description détaillée de ce niveau de plate-forme informatique est présentée à **Annexe D : Plate-forme des calculs des commandes glissantes**.

6.2 Commande glissante et les réponses temporelles

En théorie, grâce aux trajectoires d'état qui sont présentées sous forme analytique (voir Éqns. 5.23 et 5.39), la commande glissante peut être appliquée sans retard. Tandis que les trajectoires d'état sont présentées sous forme analytique, le temps retard pourrait être compensé en faisant la prédiction des changements des paramètres du régulateur du système lorsqu'il arrivera des perturbations. Cependant, dans la pratique, il faut choisir une période d'échantillonnage pour construire l'algorithme de cette commande. Cette période doit être plus petite que le temps retard lorsqu'on applique la commande par mode glissant dans le plan de phase. Par conséquent, le remplacement du plan de phase par le plan d'état est nécessaire et intéressant.

Voici l'exemple d'application de la commande glissante :

- 1) Les paramètres du modèle sont : $K_p = -0,73$, $K_i = -10,81$. Ayant ceux-ci sur le plan de phase, on remarque la présence la famille de trajectoires en forme de col où ses asymptotes sont :

$$x_2(t) = 2x_1(t); x_2(t) = -2x_1(t)$$

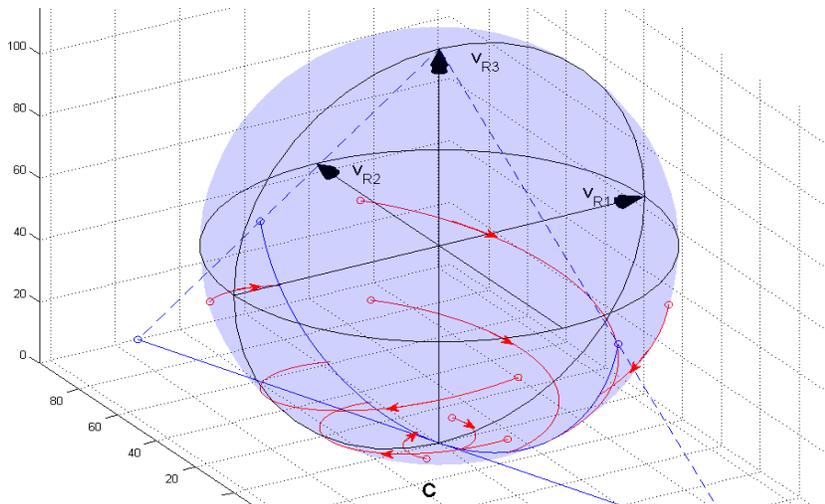
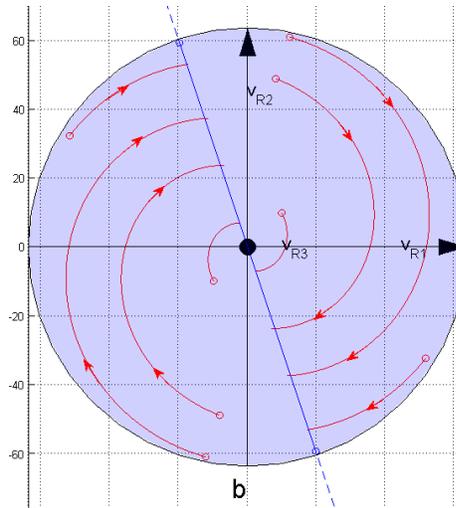
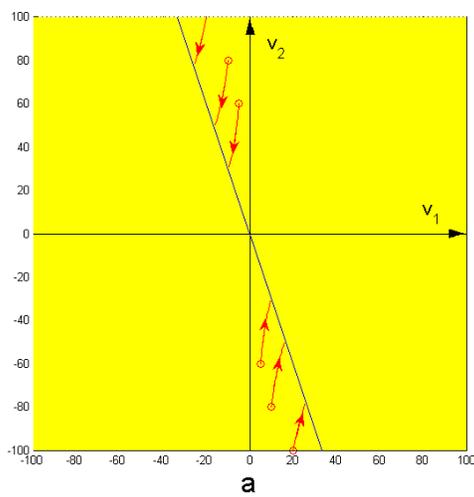
- 2) Les paramètres du modèle sont : $K_p = 1$, $K_i = 3$. Pour lesquels sur le plan de phase on a la famille de trajectoires en forme de foyer.

Ensuite, on transforme les conditions initiales et les trajectoires du système sur la sphère de **Riemann** pour définir la commande glissante (voir Chap. 5.4.1, Fig. 5.4.4). Le rayon de la sphère est défini comme $R = 50$:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 65,9 \\ 14,24 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{v}_1(t) \\ \tilde{v}_2(t) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 35,3+66,15j \\ 35,3-66,15j \end{bmatrix} \Rightarrow \begin{bmatrix} v_{R1}(t) \\ v_{R2}(t) \\ v_{R3}(t) \end{bmatrix} = \begin{bmatrix} 22,60 \\ -42,34 \\ 35,25 \end{bmatrix}$$

Il faut définir une ligne commutative dans le plan d'état et trouver sa projection sur la sphère de *Riemann*. Sur cette sphère, la ligne commutative devient la courbe commutative. Alors, si la ligne commutative est définie comme : $v_2(t) = -3v_1(t)$, on peut trouver les convergences des trajectoires d'état, voir figure 6.2.1 :



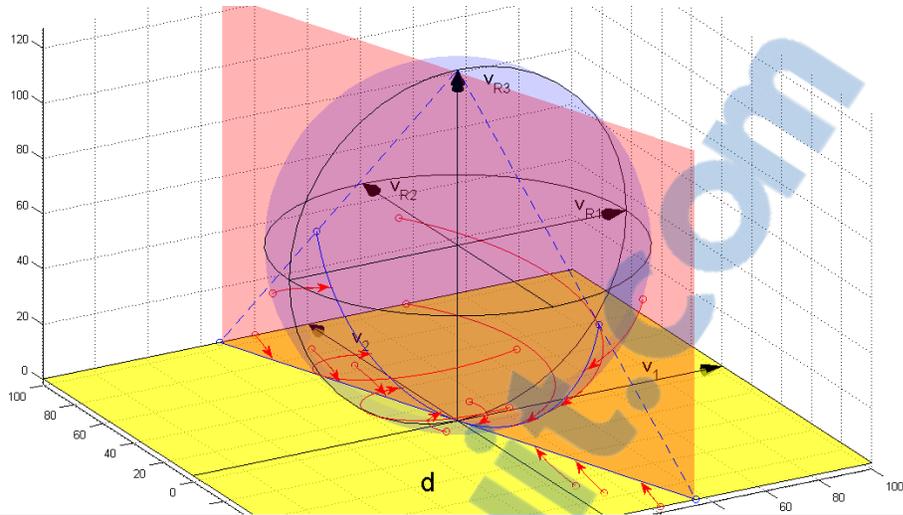


Figure 6.2.1 Démonstration des convergences de trajectoires, a) : Convergence des trajectoires hyperboles dans le plan d'état, b) : Projection la sphère de *Riemann* sur la surface $[v_{R1}, v_{R2}]$, c) : Convergence des trajectoires spirales sur la sphère de *Riemann*, d) : Construction de l'hypersphère commutative

Selon la figure 6.2.1 a) et c), les trajectoires en forme de col se dirigent vers la ligne commutative dans le plan d'état et les trajectoires en forme de foyer se dirigent vers la courbe commutative sur la sphère de *Riemann*. Alors, si on place la sphère de *Riemann* au-dessus du plan d'état avec le pôle sud de la sphère sur le point d'origine, on construit la base de la commande glissante. Dans ce cas, une hypersphère commutative peut être construite (voir Fig.6.2.1 d) surface rouge) afin de construire la commande glissante (voir Fig. 6.2.2) :

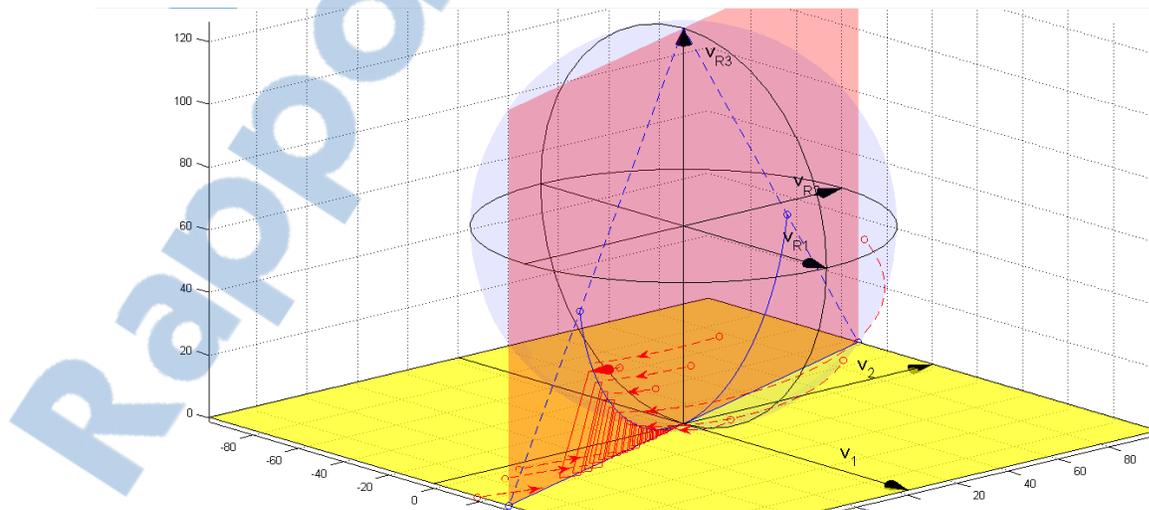


Figure 6.2.2 Trajectoire en zigzag d'une commande glissante, où : la surface jaune est le plan d'état, la sphère bleue est la sphère de *Riemann*

Alors, la surface rouge qui contient la ligne commutative (ligne bleue sur la surface jaune) et la courbe commutative (courbe bleue sur la sphère bleue) est l'hypersphère commutative. Les trajectoires pointillées rouges représentent la situation des convergences des trajectoires, et la trajectoire continue rouge en zigzag représente la trajectoire de la commande glissante.

Prenons maintenant l'agrandissement de la figure 6.2.2 pour voir mieux la trajectoire de la commande glissante :

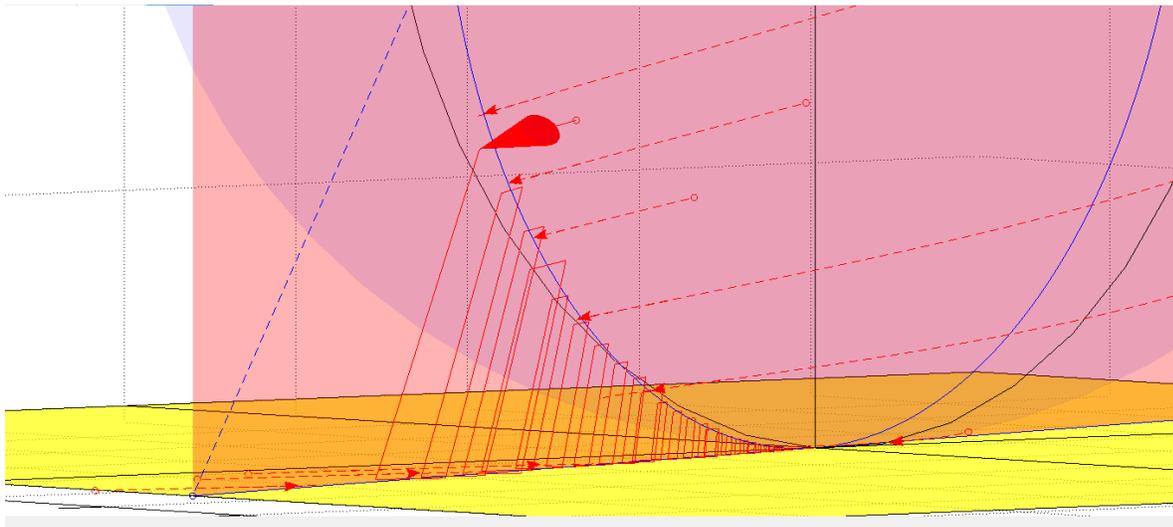


Figure 6.2.3 Agrandissement de cette trajectoire

Lorsque la trajectoire dans la sphère de *Riemann* traverse la courbe commutative, le programme obtenu par la transformation (voir **Annexe D**) modifie les paramètres du modèle tel que prescrit par le plan d'état. Au contraire, lorsque la trajectoire dans le plan d'état traverse la ligne commutative, le programme obtenu par la transformation (voir **Annexe D**) modifie les paramètres du modèle tel que prescrit par la sphère de *Riemann*. Alors, la trajectoire de la commande glissante en zigzag se dirige jusqu'au point d'origine.

Avec l'application de la commande glissante, on a obtenu la réponse temporelle :

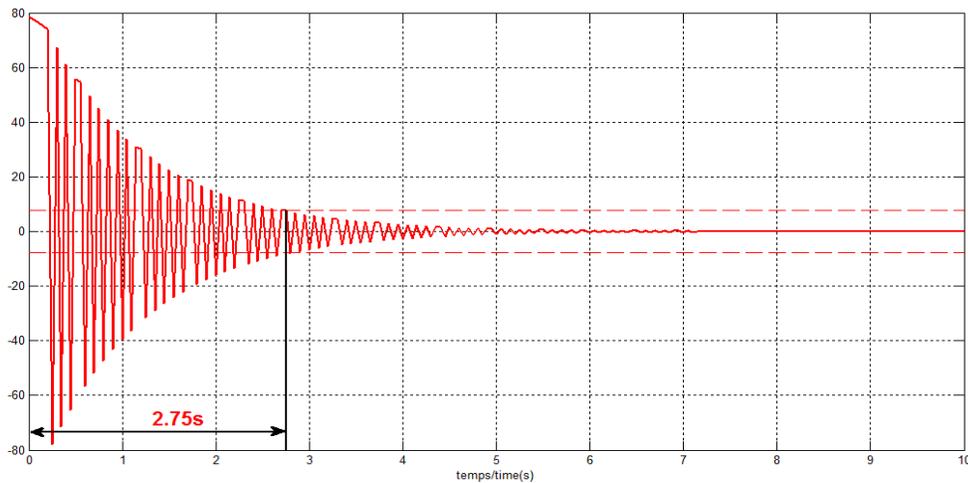


Figure 6.2.4 Réponse temporelle obtenue par l'application de la commande glissante

Selon la figure 6.2.4, les réponses temporelles se rapprochent de $\pm 10\%$ son régime définitif, c'est-à-dire que le temps de la réponse temporelle est $t_r = 2,75s$. On peut trouver que le temps de réponse temporelle diminue avec l'application de cette commande.

Assurément, on trouve qu'il existe des limites pour déplacer la ligne commutative dans le plan d'état : D'un côté, dépendamment du choix de la période d'échantillonnage, il existe une limite plus proche de la coordonnée $v_2(t)$.

De l'autre côté, en minimisant le temps de réponse temporelle, une autre limite peut être trouvée qui se rapproche de la coordonnée $v_1(t)$.

Alors, supposons maintenant, que la période d'échantillonnage est définie $t_e = 0,05s$, le temps de réponse temporelle mesuré doit être moins que $t_m = 6,84s$ (voir Chap. 4, Fig. 4.3.2). Dans ce cas, les limites de déplacement de la ligne commutative sont trouvées (voir ci-dessous) :

$$v_2(t) = -21v_1(t) \text{ (Limite rapprochée à la coordonnée } v_2(t) \text{)}$$

$$v_2(t) = -1,4v_1(t) \text{ (Limite rapprochée à la coordonnée } v_1(t) \text{)}$$

Ensuite, supposons que la période d'échantillonnage est définie comme : $t_e = 0,10s$, autres limites sont trouvées :

$$v_2(t) = -10v_1(t) \text{ (Limite rapprochée à la coordonnée } v_2(t) \text{)}$$

$$v_2(t) = -1,5v_1(t) \text{ (Limite rapprochée à la coordonnée } v_1(t) \text{)}$$

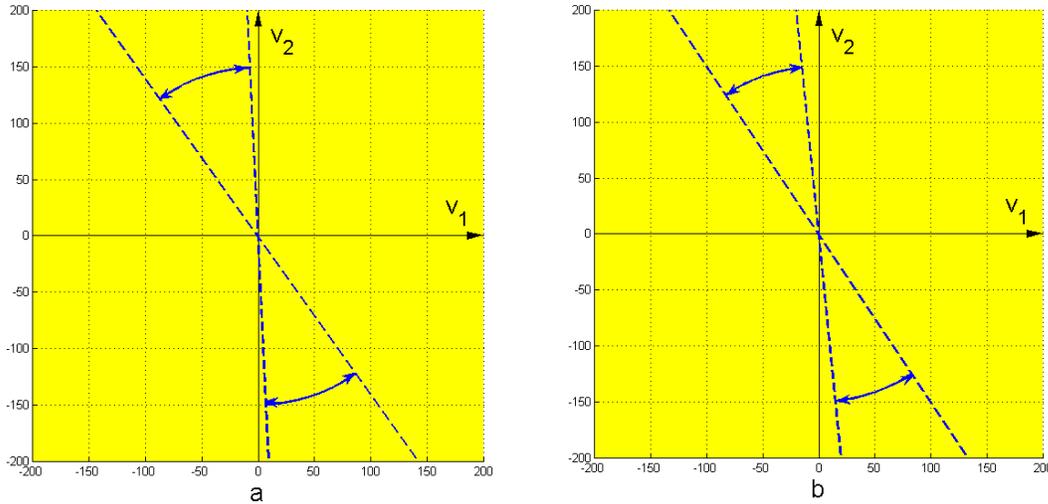


Figure 6.2.5 Limites de placement de la ligne commutative dans le plan d'état, a) : Période d'échantillonnage $t_e = 0,05s$, b) : Période d'échantillonnage $t_e = 0,10s$

Selon la figure 6.2.5, le choix de la période d'échantillonnage influence sensiblement l'intervalle des limites. Cependant, grâce à la période d'échantillonnage qui est le plus petit que le temps retard qui se trouve dans le plan de phase, la ligne commutative choisie se rapproche de la coordonnée $v_2(t)$. Donc, l'intervalle de déplacement de la ligne commutative est plus grand que celui dans le plan de phase.

Maintenant, on compare l'effet de la diminution du temps de la réponse temporelle par l'application de la commande par mode glissant dans le plan de phase avec l'effet par l'application de la commande glissante dans le plan d'état. Pour cela, on transforme les lignes commutatives qu'on a proposées dans le plan de phase au plan d'état. On a déjà présenté l'utilisation de ces lignes commutatives pour construire la commande par mode glissant (voir Chap. 4.5). Alors, la matrice modale (voir Chap.5.3 Éqn.5.17) est utilisée pour faire cette transformation :

$$\begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = M^{-1} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad (6.1)$$

Dans ce cas, les propositions de la ligne commutative dans le plan d'état sont trouvées :

$$\begin{bmatrix} x_2(t) = -0,7x_1(t) \\ x_2(t) = -x_1(t) \\ x_2(t) = -1,3x_1(t) \end{bmatrix} \Rightarrow \begin{bmatrix} v_2(t) = -2,1v_1(t) \\ v_2(t) = -3v_1(t) \\ v_2(t) = -4,7v_1(t) \end{bmatrix}$$

Les réponses temporelles des propositions des commandes par modes glissants et des propositions des commandes glissantes sont présentées ci-dessous :

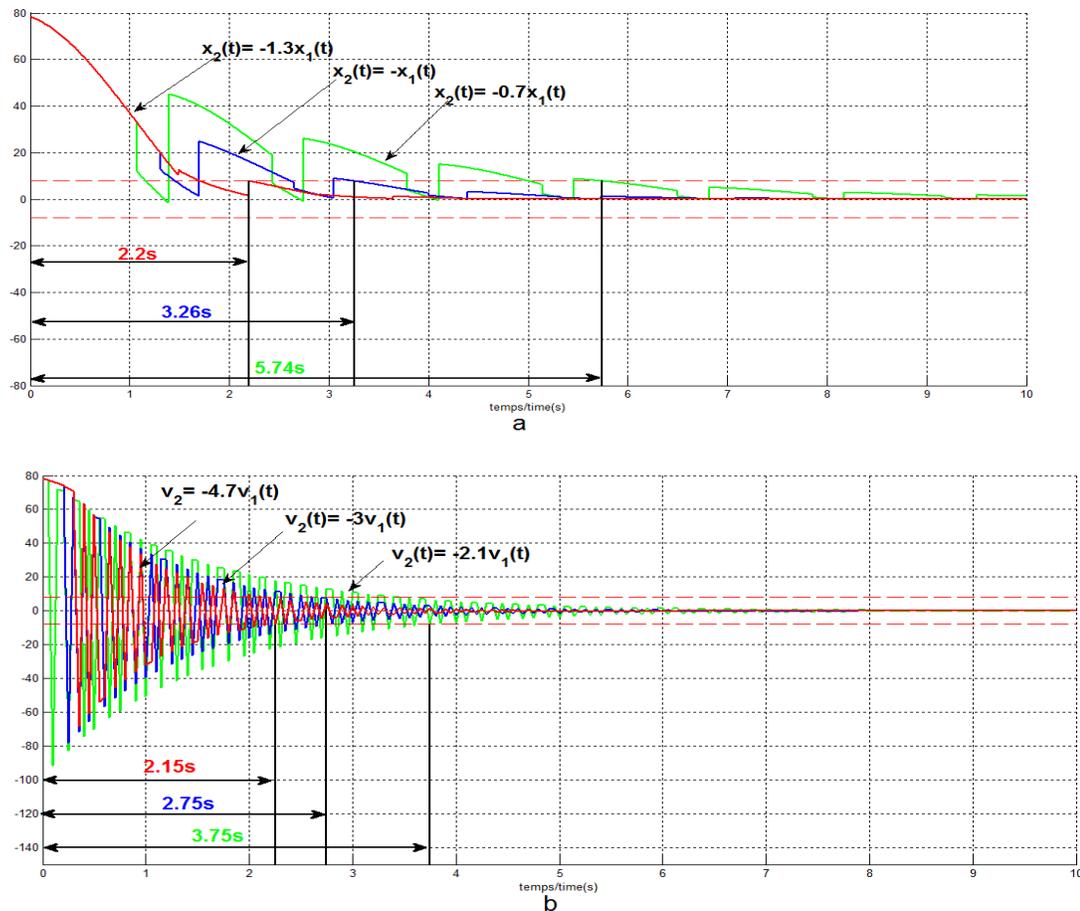


Figure 6.2.6 Réponses temporelles du système, a) : Réponses temporelles trouvées avec l'application des commandes par modes glissants, b) : Réponses temporelles trouvées avec l'application des commandes glissantes

La figure 6.2.6 a) présente les réponses temporelles d'application des commandes par modes glissants proposées dans le plan de phase. Dans ce cas, lorsque les réponses temporelles se rapprochent de

$\pm 10\%$ son régime définitif, on trouve les réponses temporelles t_r comme $(2,2s, 3,26s \text{ et } 5,74s)$. Dans l'autre côté, la figure 6.2.6 b) présente aussi les réponses temporelles d'application des commandes glissantes proposées dans le plan d'état. Alors, on trouve ces réponses temporelles t_r qui sont $(2,15s, 2,75s \text{ et } 3,75s)$. En comparant les deux groupes des temps de la réponse temporelle, on remarque que l'application des commandes glissantes diminue plus de temps de la réponse temporelle que celui de l'application des commandes par modes glissants. En effet, avec l'aide de la prédiction des trajectoires dans le plan d'état, on applique la commande glissante sans retard. Alors, l'application des commandes glissantes dans le plan d'état va minimiser le temps de la réponse temporelle du système.

6.3 Conclusion

En théorie, la commande glissante est une commande analytique, prédite explicitement dans le plan d'état qui peut être appliqué sans retard. Grâce à cette particularité, l'application de cette commande peut minimiser le temps de la réponse temporelle du système. Par conséquent, le remplacement du plan de phase par le plan d'état est nécessaire et intéressant.

Chapitre 7 Conclusions générales

Ce mémoire est consacré pour définir une nouvelle commande d'un système modélisé par le modèle à paramètres variables. Dans ce cas, la commande par mode glissant définie dans le plan de phase constitue la base de cette recherche puisqu'elle est une commande optimale. On a donc étudié une possibilité d'appliquer une commande glissante à l'aide du plan d'état. Utilisant alors la transformation de plan de phase sur le plan d'état, on a obtenu cette nouvelle commande en forme explicite. Le système choisi pour l'application de cette commande est la plate-forme mobile de soudage (PFM).

Un modèle à paramètres variables est construit pour modéliser correctement notre système étudié et ensuite faire l'application de la commande par mode glissant et la commande glissante. La structure de ce modèle est composée d'une partie connue (une fonction de transfert réduite à premier ordre), d'une partie ajustable (un correcteur PI) et une boucle de retour. Dans ce cas, les paramètres du modèle sont ajustés continuellement en fonction de l'évolution du système réel.

Par conséquent, pour voir l'effet d'application de cette commande, on construit une plate-forme informatique à trois niveaux :

1. La démonstration de l'application de la commande par mode glissant dans le plan de phase.
2. La démonstration des transformations géométriques du plan de phase au plan d'état pour trouver la base de la commande glissante.
3. La démonstration de l'application de la commande glissante dans le plan d'état.

À l'aide de cette plate-forme informatique, on peut faire la démonstration de l'application de la commande par mode glissant dans le plan de phase. Pendant l'application, on trouve qu'il existe un temps retard. Ce temps retard est causé par la restitution du point initial dans l'observateur d'état. Par conséquent, on propose de faire la transformation du plan de phase au plan d'état pour appliquer une nouvelle commande, la commande glissante.

Dans le plan d'état, les formes analytiques peuvent être trouvées pour prédire les trajectoires. Dans ce cas, les trajectoires sont sous forme analytique, réelle et complexe conjugué. Il faut remarquer que toutes les trajectoires se retrouvent dans le domaine réel pour construire la commande glissante, c'est-à-dire que les trajectoires sont réelles. Par conséquent, on utilise la sphère de *Riemann* et on fait les

transformations géométriques pour trouver la base de la commande glissante : un plan d'état (lorsque les pôles sont réels) et une sphère de *Riemann* (lorsque les pôles sont complexes conjugués). Dans ce cas, une hypersphère commutative peut être définie pour construire cette nouvelle commande.

Grâce à la prédiction des trajectoires dans le plan d'état, la commande glissante est en forme explicite. Dans ce cas, le temps retard pourrait être compensé en faisant la prédiction des changements des paramètres du régulateur du système lorsqu'il existe des perturbations. Donc, la commande glissante peut être appliquée sans retard. Avec cette particularité, la commande glissante peut minimiser le temps de la réponse temporelle. On a donc obtenu une nouvelle commande dans le plan d'état. Par conséquent, le remplacement du plan de phase par le plan d'état est nécessaire et intéressant.

Références

- [1] Frddy Mudry : « Ajustage des Paramètres des Régulateurs PID », Laboratoire d'Automatique, Département d'Électricité et Informatique, École d'ingénieurs du Canton de Vaud, 4, 2002.
- [2] Daniel Ross, Etienne Deguine et Mickaël Camus : « Asservissement par PID », rose.eu.org, 03, 2010.
- [3] K.J Åström and T. Hägglund: PID Controllers : Theory, Design and Tuning, Second Edition, Instrument Society of America, 1995.
- [4] U. Itkis : Control systems of variable structure, John Wiley& Sons, Incorporated, ISBN-10:0470150726, 1976.
- [5] Vadim I. Utkin : « Variable Structure Systems with Sliding Modes », IEEE Transaction on Automatic Control, Vol.AC-22, NO.2, 1977, pp. 212-221.
- [6] Su W C, Drakunov S V, Ozguner U et al : « Sliding mode with chattering reduction in sampled data systems », IEEE Press, 12, 1993, pp. 2452-2457.
- [7] Krupp D et Shtessel Y B : « Chattering-free sliding mode control with unmodeled dynamics », IEEE Press, 6, 1999, pp. 530-534.
- [8] Kim Y S, Han Y S et You W S : « Disturbance observer with binary control theory », IEEE Press, 6, 1996, pp. 1229-1234.
- [9] Lin F J et Chou W D : « An induction motor servo drive using sliding mode controller with genetic algorithm », Electric Power Systems Research, 2003, 64(2), pp. 93-108.
- [10] Edwards C : « A practical method for the design of sliding mode controllers using linear matrix inequalities », Automatic, 2004,40(10), pp. 1761-1769.
- [11] Beaudoin, Martin : « Pilotage d'un mécanisme mobile à la base d'une commande à structure Variable », Mémoire (M. Sc.)—Université Laval, 2003.
- [12] Stanislaw Tarasiewicz : « State space multiple leaves for linear time-varying systems and attempting sliding mode controllers » : an original approach, WSEAS, Greece, 2006, pp. 1109-2777.
- [13] Stanislaw Tarasiewicz : Rapport Technique de l'identification d'une Plate-forme mobile, Laboratoire d'Automatisation Complexe et de Mécatronique(LACM), Département de Génie Mécanique, Université Laval, pp. 1-50.
- [14] Lennart Ljung : « System identification », Département de Génie Électricité, Université Linköping, Suède.

- [15] Lennart Ljung : System Identification Toolbox User's Guide R2013b, Math Works, 2013.
- [16] Chiara Masiero : Matlab System Identification Toolbox Quickstart, Département de Génie Informatique, Université Padova, Italie, 2012.
- [17] Belatra, Karim : « L'identification en ligne et la commande par mode de glissement d'un système mécanique complexe : application, robot et plate-forme mobile », Mémoire (M. Sc.)— Université Laval, 2005.
- [18] C.B. Vishwakarma : « Order Reduction using Modified Pole Clustering and Pade Approximation », World Academy of Science, Engineering and Technology, 2011.
- [19] Maurice F. Hutton and Bernard Friedland : « Routh approximations for reducing order of Linear, time-invariant systems », IEEE Transaction on Automatic Control, vol. AC-20, No. 3, 1975.
- [20] Younseok Choo et Dongmin Kim : « Routh approximants with arbitrary, the institute of Control », Département de Génie Automatisation et Système, Corée du sud, Vol. 1, No. 1, 1999.
- [21] Y. Shamash : « Linear system reduction using Pade approximation to allow retention of dominant modes », International Journal of Control, Vol. 21, No.2, 1975, pp. 257-272.
- [22] René Husson, Claude Lung, Jean-François Aubry, Jamal Daafouz et Didier Wolf : Automatique du cahier des charges à la réalisation de systèmes, Chapitre 5, Dunod, 2007, pp. 90-99.
- [23] Stanislaw Tarasiewicz : Note de cours d'Automatique et Automatisation en Modélisation des Systèmes Asservis à l'aide des réponses temporelles, Département de Génie Mécanique, Université Laval, 2013.
- [24] Stanislaw Tarasiewicz : Note de cours de Servomécanismes et Contrôleurs en Analyse et Régulation des systèmes mécaniques, Département de Génie Mécanique, Université Laval, 2011.
- [25] Stanislaw Tarasiewicz : Note de cours en Dynamique de la commande appliquée et Asservissement, Département de Génie Mécanique, Université Laval, 2011.
- [26] Jean-Charles Gille et Marc Clique : Systèmes Linéaires Équations D'état, Éditions Eyrolles, 1990, pp. 69-71, 91-103.
- [27] Thomas Banchoff et John Wermer : Linear Algebra Through Geometry, second Editions, Springer-Verlag New York, 1992.
- [28] Derek Rowell : « Analysis and Design of Feedback Control Systems State-Space Representation of LTI Systems », 2002.
- [29] J.-Ch. Gille, P. Decaulne et M. Pélegrin : Systèmes asservis non linéaires, Dunod, 1988, Chapitre 1, pp. 15-21, Chapitre 5 pp. 109-126, Chapitre 10, pp. 7-10.
- [30] Friedland et Bernard : « Control system design: an introduction to state-space methods »,

McGraw-Hill, 1986.

- [31] Katsuhiko Ogata : Moden Control Engineering, Chapitre 14, Prentice-Hall, Englewood Cliffs, N. J, 1970, pp. 563-576, 663-678.
- [32] B. Wayne et Bequette : Process Dynamics Modeling Analysis and Simulation, Prentice Hall PTR, 1998, pp. 304-314, 315-316.
- [33] B. L. Deekshatulu et I. S. N.Murthy : « Generalized Isocline Method of Plotting Phase-Plane Trajectoires », IEEE Transaction on Automatic Control, 1965, pp. 90-92.
- [34] Stanislaw Tarasiewicz : Note de cours d'un sujet spécial : Construction d'un simulateur numérique/graphique pour analyser des trajectoires d'état dans un plan à feuilles multiples, Département de Génie Mécanique, Université Laval, 2013.
- [35] P.Naslin : Théorie de la commande et conduite optimale, Chapitre 6, Dunod, 1969, pp. 112 – 119.
- [36] Guy Gauthier : Notre de cours : Plan de phase, Département de génie de la production automatisée, École de technologie supérieure, 2012.
- [37] GS Guide : Getting Started with Simulink, R2013b, Math Works, 2013.
- [38] Prix Roberval : Analyse Numérique pour Ingénieurs deuxième édition, École Polytechnique de Montréal, 2001.
- [39] Frédéric Rotella : Note de cours : Observation, École Nationale d'Ingénieurs de Tarbes.
- [40] Murdoch : « Observer design for a linear functional of the state vector », IEEE Transaction on Automatic Control, AC-18, 1973, pp. 308-310.
- [41] Vadim Utkin : « Sliding Mode Control, Control Systems », Robotics and Automation, Vol. XIII, EOLSS.
- [42] Donald E, Kirk : Optimal control theory: an introduction, Prentice-Hall, 1970.
- [43] Stanislaw Tarasiewicz : Note de cours de la Commande Optimale, Sujet : Définition d'une Commande glissante à 3D, Département de Génie Mécanique, Université Laval, 2012.
- [44] Stanislaw Tarasiewicz : Cours d'Automatique et automatisation, Rapport numéro 4, Département de Génie Mécanique, Université Laval, 14, 12, 2011.
- [45] Thomas Banchoff and John Wermer : Linear Algebra through Geometry second Edition, Springer-Verlag New York, 1992.
- [46] Stanislaw Tarasiewicz : « Transformations : Cas, Valeurs propres réels et complexes », Rapport Théorique, Département de Génie Mécanique, Université Laval, 07, 2014.
- [47] Noh Seong Cheol : « Introduction to Riemann Surface », Physical Mathematics Conference, 2010.

- [48] Stanislaw Tarasiewicz : Rapport théorique sur la transformation complexe. Département de Génie Mécanique, Université Laval, 2014.
- [49] Dr C. Teleman : « Riemann Surfaces », Lent Terme, 2003.
- [50] J. Sauloy : « Initiation aux surfaces de Riemann », Institut mathématique de Toulouse et U.F.R. M.I.G, Université Paul Sabatier, 2011.

Annexe A : Plate-forme des calculs d'approximation des réponses temporelles

Dans les travaux d'ingénierie, particulièrement dans les applications en temps réel, on a souvent besoin d'effectuer des calculs par ordinateur de façon quasi instantanée. La rapidité des calculs ainsi que la nécessité de visualiser les variables d'état d'un système nous amène à considérer l'application de la théorie de la réduction dans le modèle mathématique d'un système. Le modèle réduit possède plusieurs avantages évidents. Il est facile à manipuler et il permet de représenter fidèlement la dynamique du système. En outre, un modèle réduit à paramètres ajustables permet de représenter adéquatement, sur un intervalle de temps restreint, des systèmes complexes. Dans ce cas, cette plate-forme est construite pour faire l'identification expérimentale et la visualisation des réponses temporelles à l'aide des modèles approximatifs. Les modèles approximatifs sont les modèles réduits et le modèle à paramètres variables.

La plate-forme des calculs d'approximation des réponses temporelles est construite dans le logiciel *Matlab*. Elle se divise en 6 sections comme la montre dans la figure ci-dessous :

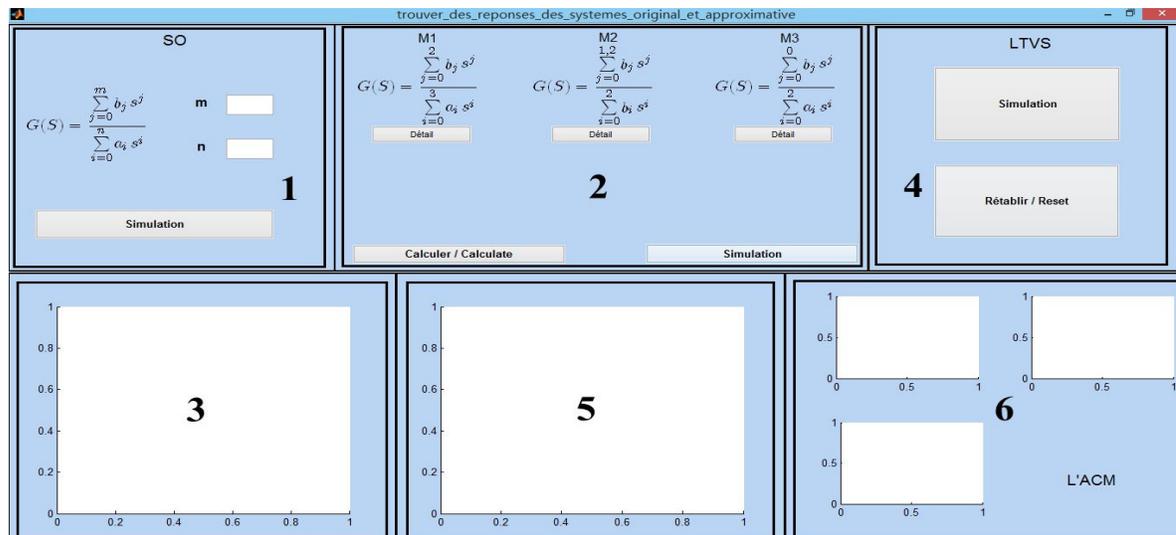


Figure A.1 Interface graphique de la plate-forme des calculs d'approximation des réponses temporelles du système.

Description :

- Section [1] : Cette section sert à spécifier le système d'ordre supérieur avec les définitions des valeurs des paramètres de la fonction de transfert du système. Si on entre les valeurs d'ordre du dénominateur et le numérateur de la fonction de transfert dans les cases blanc, la plate-forme va ouvrir deux fenêtres pour faire les définitions des paramètres du système (voir Fig. A.2). Il faut que la définition d'ordre de dénominateur soit plus grande que celui de numérateur.
- Section [2] : Cette section consacrée aux calculs sert à réduire l'ordre du système et à trouver les modèles d'ordre réduits. Les méthodes de réduction utilisées sont la méthode d'approximation de Routh (M1), la méthode Schur (M2) et la méthode d'« Optimal Hankel norme approximation » (M3).
- Section [3] : Cette section est dédiée à la visualisation de la réponse temporelle du système d'ordre supérieur et les réponses temporelles des modèles d'ordre réduits pour voir l'effet d'approximation.
- Section [4] : Cette section est consacrée à la simulation de l'application du modèle à paramètres variables.
- Section [5] : Cette section est dédiée à la visualisation de la réponse temporelle du système d'ordre supérieur et la réponse temporelle d'application du modèle à paramètres variables pour voir l'effet d'approximation.
- Section [6] : Cette section est dédiée à la visualisation des changements des valeurs des paramètres du correcteur du modèle à paramètres variables.

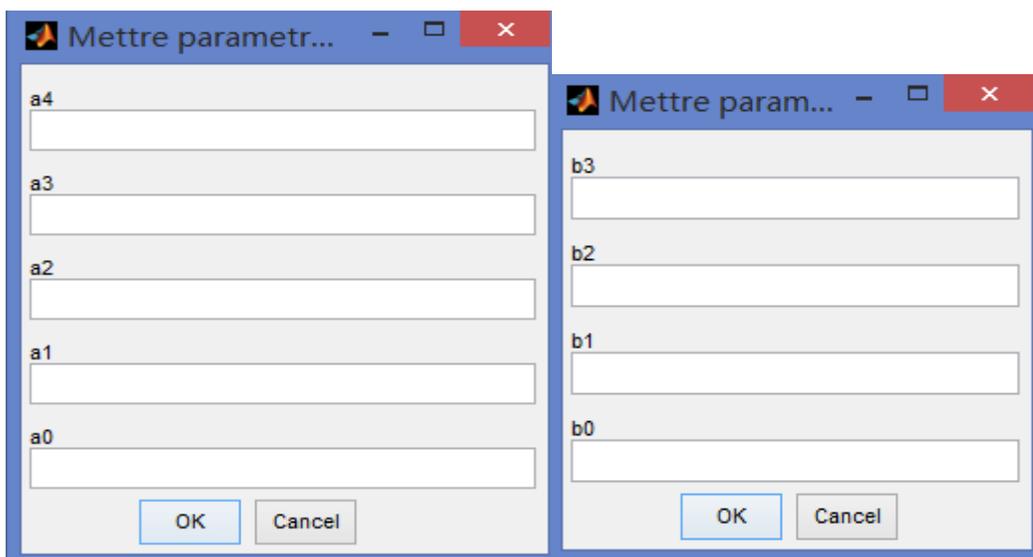


Figure A.2 Définitions les paramètres du système d'ordre supérieur ($m=3$, $n=4$).

Pour utiliser cette plate-forme, il faut effectuer les étapes suivantes :

- Définir les paramètres de la fonction de transfert du système supérieur, c'est-à-dire que l'utilisateur doit définir les valeurs des paramètres de dénominateur et numérateur de la fonction de transfert.
- Cliquer sur le bouton **Calculer / Calculate** dans la section [2] pour lancer les programmes de calcul. Obtenir les modèles réduits. Cliquer sur le bouton **Simulation** dans la section [2] pour tracer les réponses temporelles approximées dans la section [3].
- Cliquer sur le bouton **Simulation** dans la section [4] pour faire la simulation du modèle à paramètres variables et ensuite tracer la réponse temporelle approximée dans la section [5].

Un exemple de la simulation de cette plate-forme est présenté dans la figure A.3 ci-dessous :

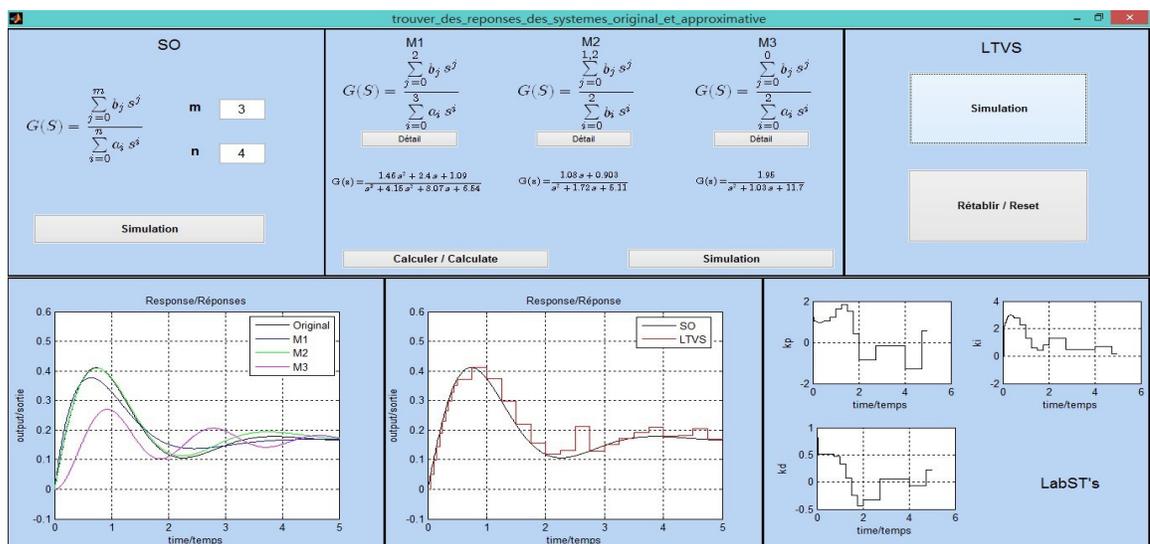


Figure A.3 Visualisation d'une simulation des calculs d'approximation des réponses temporelles.

Grâce aux paramètres ajustables du correcteur, la proposition d'approximation trouvée par le modèle à paramètres variables suit correctement de la réponse temporelle originale. La même méthode a été utilisée pour faire la construction du modèle à paramètres variables de notre système étudié (voir Chap. 2.4, Éqn.2.30).

Voici une partie des programmations de la réduction et d'identification se font à l'aide du logiciel **Matlab** :

`%méthodes de réduire l'ordre du système`

```

%Routh méthode
if n>=4;
    digits(4);
%la fonction de transfert du système originale
    tfOriginal=tf(sb,sa)
    nr=3;n0=length(sa);n1=length(sb);
    a1=sa(end:-1:1);b1=[sb(end:-1:1) zeros(1,n0-n1-1)];
    for k=1:n0-1,
        k1=k+2;alpha(k)=a1(k)/a1(k+1);beta(k)=b1(k)/a1(k+1);
        for i=k1:2:n0-1,
            a1(i)=a1(i)-alpha(k)*a1(i+1);b1(i)=b1(i)-beta(k)*a1(i+1);
        end
    end
    nn=[];dd=[1];nn1=beta(1);dd1=[alpha(1),1];nred=nn1;dred=dd1;
    for i=2:nr,
        nred=[alpha(i)*nn1,beta(i)];dred=[alpha(i)*dd1,0];
        n0=length(dd);n1=length(dred);nred=nred+[zeros(1,n1-n0),nn];
        dred=dred+[zeros(1,n1-n0),dd];
        nn=nn1;dd=dd1;nn1=nred;dd1=dred;
    end
    rtf=tf(nred(nr:-1:1),dred(end:-1:1));assignin('base','rtf',rtf);
    numRedure1=rtf.num{1,1};assignin('base','numRedure1',numRedure1);
    denRedure1=rtf.den{1,1};assignin('base','denRedure1',denRedure1);
%Schur méthode
    sys0=tf(sb,sa);sys0=ss(sys0);
    sys1=schmr(sys0,1,2);sys1=tf(sys1);assignin('base','ceshitf',sys1);
    numRedure2=sys1.num{1,1};assignin('base','numRedure2',numRedure2);
    denRedure2=sys1.den{1,1};assignin('base','denRedure2',denRedure2);
%opt méthode
    G_Sys=tf(sb,sa);
    r=0;k=2;
    G_r=opt_app(G_Sys,r,k,0);assignin('base','G_r',G_r);
    numRedure3=G_r.num{1,1};assignin('base','numRedure3',numRedure3);
    denRedure3=G_r.den{1,1};assignin('base','denRedure3',denRedure3);
    r=2;k=3;
    G_rr=opt_app(G_Sys,r,k,0);
    numRedure4=G_rr.num{1,1};assignin('base','numRedure4',numRedure4);
    denRedure4=G_rr.den{1,1};assignin('base','denRedure4',denRedure4);
else
%Schur méthode
    digits(4);
    sys0=tf(sb,sa);sys0=ss(sys0);
    sys1=schmr(sys0,1,2);sys1=tf(sys1);
    numRedure2=sys1.num{1,1};assignin('base','numRedure2',numRedure2);
    denRedure2=sys1.den{1,1};assignin('base','denRedure2',denRedure2);
%opt méthode
    G_Sys=tf(sb,sa);
    r=0;k=2;
    G_r=opt_app(G_Sys,r,k,0);assignin('base','G_r',G_r);
    numRedure3=G_r.num{1,1};assignin('base','numRedure3',numRedure3);
    denRedure3=G_r.den{1,1};assignin('base','denRedure3',denRedure3);
    r=2;k=3;
    G_rr=opt_app(G_Sys,r,k,0);
    numRedure4=G_rr.num{1,1};assignin('base','numRedure4',numRedure4);
    denRedure4=G_rr.den{1,1};assignin('base','denRedure4',denRedure4);
end

```

%identification de la réponse temporelle par le modèle à paramètres variable du troisième ordre dans la plateforme

```

if abs([Ydata(i1+1,1) — PY(end,1)]./max[YS0])<=0.3&&abs([Ydata(i1+1,1) — PY(end,1)]./max[YS0])>0.05;
    TSY0=t0(1,end):0.005:(t0[1,end]+te);
    USYS0=ut.*ones(length(TSY0),1);
    [YSYS0,TSYS0,XSYS0]=lsim(SYS0,USYS0,TSYS0,SYS0X(end,:));
    iddataSYS0=iddata(YSYS0,USYS0,0.005);
    par=[kpp(1,end);kii(1,end);kdd(1,end)];
    aux={};Ts=0;
    PIDmodel=idgrey(« fzfunc »,par,'c',aux,Ts);assignin('base','PIDmodel',PIDmodel);
    opt=greystOptions;
    opt.Focus='stability';
    opt.SearchOption.MaxIter=20;
    PIDVariablemodel=greyst(iddataSYS0,PIDmodel,opt);
    kpp(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(1,1);
    kii(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(2,1);
    kdd(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(3,1);
    a0(1,end+1)=b20.*kii(1,end);
    a1(1,end+1)=(a20+b20.*kpp(1,end));
    a2(1,end+1)=(a21+b20.*kdd(1,end));
    b0(1,end+1)=b20.*kii(1,end);
    b1(1,end+1)=b20.*kpp(1,end);
    b2(1,end+1)=b20.*kdd(1,end);
    APID(3,1)=-1*a0(1,end);
    APID(3,2)=-1*a1(1,end);
    APID(3,3)=-1*a2(1,end);
    CPID(1,1)=b0(1,end);
    CPID(1,2)=b1(1,end);
    CPID(1,3)=b2(1,end);
    PSYS=ss(APID,BPID,CPID,DPID);
    [PY0,PT0,PX0]=lsim(PSYS,USYS0,TSYS0,PX(end,:));
    %mise a jour
    t0(1,end+1)=t0(1,end)+te;
    PX=[PX;PX0(end,:)];
    PY=[PY;PY0(end,1)];
    SYS0X=[SYS0X;XSYS0(end,:)];
    SO0=[SO0;YSYS0(end,1)];

```

%identification de la réponse temporelle par le modèle à paramètres variable du second ordre de la PFM (voir Chap.2.2 Éqn. 2.30)

%la fonction de transfert du système d'ordre supérieur

```

a=[2.6930 9.3770 13.1300 6.0680 1.0000];
b=[3.7780 0.9405 -1.4130];
n=4;m=2;ts=10;te=0.2;tx=10;tmin=0.2;ut=1;
A=eye(n+1);
A(1,:)=[];A(:,end)=[];
for i1=1:n;
    A(end,i1)=-1*a(1,i1);
end
B=zeros(n,1);
B(n,1)=1;
C=zeros(1,n);
if m<n;
for i2=1:m+1;
    C(1,i2)=b(1,i2);
end
end

```

```

D=0;
SYS0=ss(A,B,C,D);
%simulation pour trouver la réponse temporelle mesurée
T0=0:0.2:ts;
T1=0:te:ts;
U0=ut*ones(length(T0),1);%
U1=ut*ones(length(T1),1);%
X0=zeros(1,n);
hold on
[YS0,TS0,XS0]=lsim(SYS0,U0,T0,X0);
plot(TS0,YS0,'r');
hold on
YS0=awgn(YS0,30,'measured');
plot(TS0,YS0,'k');
hold on
grid on
[Ydata0,Tdata0,Xdata0]=lsim(SYS0,U1,T1,X0);
if (ts./te)==fix(ts./te);
Ydata=Ydata0;
Tdata=Tdata0;
Xdata=Xdata0;
else
Ydata=[Ydata0;YS0(end,1)];
Tdata=[Tdata0;TS0(end,1)];
Xdata=[Xdata0;XS0(end,:)];
end
%construire le modèle à paramètres variables de la PFM
global PX a20 b20;
a20=0.27;b20=0.37;
PX=zeros(1,2);
PY=0;
t0=0;S00=0;
SYS0X=zeros(1,n);
TSYS0=0:0.01:tx;
USYS0=ut.*ones(length(TSYS0),1);
[YSYS0,TSYS0,XSYS0]=lsim(SYS0,USYS0,TSYS0,SYS0X(end,:));
%comparaison la réponse temporelle pour faire l'identification des paramètres du correcteur
iddataSYS0=iddata(YSYS0,USYS0,0.005); %% 0.005
par=[((0.909-a20)/b20+(-0.258/b20))/2;(0.356+0.488)/(2*b20)];
aux={};Ts=0;
% utilise la sub fonction fzzfunc
PIDmodel=idgrey(« fzzfunc »,par,'c',aux,Ts);assignin('base','PIDmodel',PIDmodel);
%trouver les paramètres du correcteur (initiales valeurs)
zpm=zpk(PIDmodel)
opt=greyestOptions;
opt.Focus='stability';
opt.SearchOption.MaxIter=300;
PIDVariablemodel=greyest(iddataSYS0,PIDmodel,opt);
kpo=PIDVariablemodel.report.Parameters.ParVector(1,1);
kio=PIDVariablemodel.report.Parameters.ParVector(2,1);
kii=kio;
kpp=kpo;
% calculer les paramètres du modèle
a0=b20.*kii;
a1=(a20+b20.*kpp);
b0=b20.*kii;

```

```

b1=b20.*kpp;
APID=eye(3);
APID(1,:)=[];
APID(:,end)=[];
BPID=zeros(2,1);
BPID(2,1)=1;
CPID=zeros(1,2);
DPID=0;
% approximation la réponse temporelle et calculer le changement des paramètres du correcteur
for i1=1:(length(Ydata)-1);
    if
        abs([Ydata(i1+1,1) — PY(end,1)]./max[YS0])<=0.3&&abs([Ydata(i1+1,1) — PY(end,1)]./max[YS0])>0.05;
            TSYS0=t0(1,end):0.005:(t0[1,end]+te);%%%%%% 0.005
            USYS0=ut.*ones(length(TSYS0),1);
            [YSYS0,TSYS0,XSYS0]=lsim(SYS0,USYS0,TSYS0,YSYS0,YS0X(end,:));
            iddataSYS0=iddata(YSYS0,USYS0,0.005);%%%%%%%%%% %0.005
            par=[kpp(1,end);kii(1,end)];
            aux={};
            Ts=0;
            PIDmodel=idgrey(« fzzfunc »,par,'c',aux,Ts);assignin('base','PIDmodel',PIDmodel);
            opt=greyestOptions;opt.Focus='stability';
            opt.SearchOption.MaxIter=20;
            PIDVariablemodel=greyest(iddataSYS0,PIDmodel,opt);
            kpp(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(1,1);
            kii(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(2,1);
            a0(1,end+1)=b20.*kii(1,end);
            a1(1,end+1)=(a20+b20.*kpp(1,end));
            b0(1,end+1)=b20.*kii(1,end);
            b1(1,end+1)=b20.*kpp(1,end);
            APID(2,1)=-1*a0(1,end);
            APID(2,2)=-1*a1(1,end);
            CPID(1,1)=b0(1,end);
            CPID(1,2)=b1(1,end);
            PSYS=ss(APID,BPID,CPID,DPID);
            [PY0,PT0,PX0]=lsim(PSYS,USYS0,TSYS0,PX(end,:));
            %mise a jour et trouver l'approximation de la réponse
            t0(1,end+1)=t0(1,end)+te;
            PX=[PX;PX0(end,:)];
            PY=[PY;PY0(end,1)];
            SYS0X=[SYS0X;XSYS0(end,:)];
            SO0=[SO0;YSYS0(end,1)];
    elseif abs([Ydata(i1+1,1) — PY(end,1)]./max[YS0])>0.3;
        for j1=1:(te/tmin);
            TSYS0=t0(1,end):0.005:(t0(1,end)+tmin);%%%%%%%%%% 0.005
            USYS0=ut.*ones(length(TSYS0),1);
            [YSYS0,TSYS0,XSYS0]=lsim(SYS0,USYS0,TSYS0,YSYS0,YS0X(end,:));
            iddataSYS0=iddata(YSYS0,USYS0,0.005);%%%%%%%%%% 0.005
            par=[kpp(1,end);kii(1,end)];
            aux={};Ts=0;
            PIDmodel=idgrey(« fzzfunc »,par,'c',aux,Ts);
            opt=greyestOptions;opt.Focus='stability';
            opt.SearchOption.MaxIter=20;
            PIDVariablemodel=greyest(iddataSYS0,PIDmodel,opt);
            kpp(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(1,1);
            kii(1,end+1)=PIDVariablemodel.report.Parameters.ParVector(2,1);
            a0(1,end+1)=b20.*kii(1,end);

```

```

a1(1,end+1)=(a20+b20.*kpp(1,end));
b0(1,end+1)=b20.*kii(1,end);
b1(1,end+1)=b20.*kpp(1,end);
APID(2,1)=-1*a0(1,end);
APID(2,2)=-1*a1(1,end);
CPID(1,1)=b0(1,end);
CPID(1,2)=b1(1,end);
PSYS=ss(APID,BPID,CPID,DPID);
[PY0,PT0,PX0]=lsim(PSYS,USYS0,TSYS0,PX(end,:));
%mise a jour et trouver l'approximation de la réponse
t0(1,end+1)=t0(1,end)+tmin;
PX=[PX;PX0(end,:)];
PY=[PY;PY0(end,1)];
SYS0X=[SYS0X;XSYS0(end,:)];
SO0=[SO0;YSYS0(end,1)];
end
elseif abs([Ydata(i1+1,1)— PY(end,1)]./max[YS0])<0.05;
TSYS0=t0(1,end):0.005:(t0[1,end]+te);
USYS0=ut.*ones(length(TSYS0),1);
[YSYS0,TSYS0,XSYS0]=lsim(SYS0,USYS0,TSYS0,SYS0X(end,:));
YSYS0=awgn(YSYS0,20,'measured');
kpp(1,end+1)=kpp(1,end);
kii(1,end+1)=kii(1,end);
a0(1,end+1)=b20.*kii(1,end);
a1(1,end+1)=(a20+b20.*kpp(1,end));
b0(1,end+1)=b20.*kii(1,end);
b1(1,end+1)=b20.*kpp(1,end);
APID(2,1)=-1*a0(1,end);
APID(2,2)=-1*a1(1,end);
CPID(1,1)=b0(1,end);
CPID(1,2)=b1(1,end);
PSYS=ss(APID,BPID,CPID,DPID);
[PY0,PT0,PX0]=lsim(PSYS,USYS0,TSYS0,PX(end,:));
%mise a jour et trouver l'approximation de la réponse
t0(1,end+1)=t0(1,end)+te;
PX=[PX;PX0(end,:)];
PY=[PY;PY0(end,1)];
SYS0X=[SYS0X;XSYS0(end,:)];
SO0=[SO0;YSYS0(end,1)];
end
end
%sub fonction fzzfunc
function [A,B,C,D,K,x0] = fzzfunc(par,Ts);
global PX a20 b20;
kp = par(1);
ki = par(2);
A = [0,1;-1*b20*ki,-1*(a20+b20*kp)];
B = [0;1];
C = [b20*ki,b20*kp];
D = zeros(1,1);
K = zeros(2,1);
x0 =[PX(end,1);PX(end,2)];
end

```

Annexe B : Plate-forme des calculs des commandes par modes glissants

Cette plate-forme est le premier niveau de la plate-forme informatique où les familles de trajectoire de phase en forme de col et en forme de foyer sont utilisées pour construire la commande par mode glissant. On construit cette plate-forme pour obtenir des informations et pour faire la visualisation d'application de la commande par mode glissant.

La plate-forme des calculs des commandes par modes glissants est construite dans le logiciel *Matlab*. Elle se divise en 10 sections comme la montre dans la figure ci-dessous :

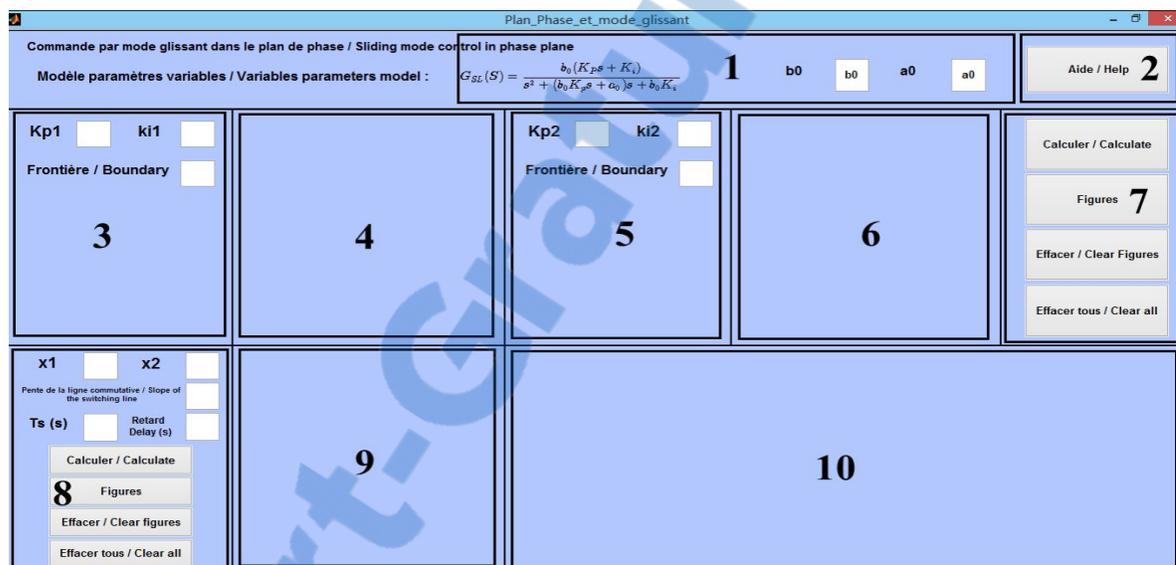


Figure B.1 Interface graphique de la plate-forme des calculs des commandes par modes glissants.

Description :

- Section [1] : Cette section sert à spécifier les paramètres du modèle à paramètres variables : a_0, b_0 .
- Section [2] : Cette section sert à ouvrir une fenêtre qui présente les informations d'aide de cette plate-forme. La fenêtre d'aide est présentée dans la figure B.2.
- Section [3] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i et à définir la frontière de la figure pour tracer la famille de trajectoires en forme de col dans la section [4]. Il existe une restriction pour définir les valeurs des paramètres K_p, K_i où les pôles du système trouvés par les définitions des paramètres K_p, K_i sont réels. Si l'utilisateur essaye de définir

les valeurs des paramètres pour un système qui ont les pôles complexes conjugués, il reçoit le message d'erreur présentée dans la figure B.3.

- Section [4] : Cette section est dédiée à la visualisation de la famille de trajectoires en forme de col dans le plan de phase.
- Section [5] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i et à définir la frontière de la figure pour tracer la famille de trajectoires en forme de foyer dans la section [6]. Il existe aussi une restriction pour définir des valeurs des paramètres K_p, K_i où les pôles du système trouvés par les définitions des paramètres K_p, K_i sont complexes conjugués. Si l'utilisateur essaye de définir les valeurs des paramètres pour un système qui ont des pôles réels, il reçoit le message d'erreur présentée dans la figure B.4.
- Section [6] : Cette section est dédiée à la visualisation de la famille de trajectoires en forme de foyer dans le plan de phase.
- Section [7] : Cette section est consacrée aux calculs pour trouver la famille de trajectoires en forme de col et la famille de trajectoires en forme de foyer. Après les calculs, ces deux familles de trajectoires sont tracées dans les sections [4] et [6] respectivement.
- Section [8] : Cette section sert à spécifier les paramètres de la démonstration de l'application de la commande par mode glissant : les conditions initiales $(x_1(0), x_2(0))$ dans le plan de phase, la ligne commutative, le temps de simulation $T_s(s)$ et le temps retard.
- Section [9] : Cette section est dédiée à la visualisation des trajectoires des commandes par modes glissants. La démonstration est présentée dans plusieurs possibilités : la commande par mode glissant d'un processus idéal, la commande idéale appliquée au système étudié et la commande par mode glissant appliquée au système étudié.
- Section [10] : Cette section est dédiée à la visualisation des réponses temporelles des commandes par modes glissants proposées dans la section [9].

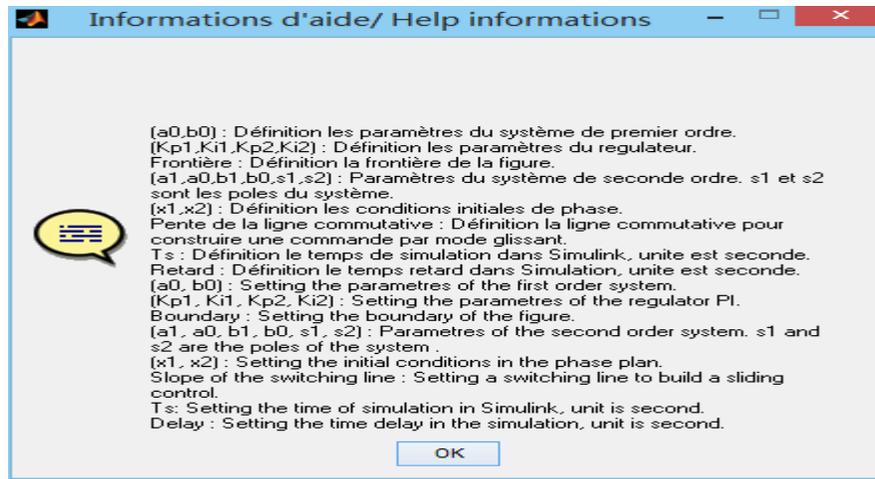


Figure B.2 Fenêtre d'aide.



Figure B.3 Message d'erreur No.1.

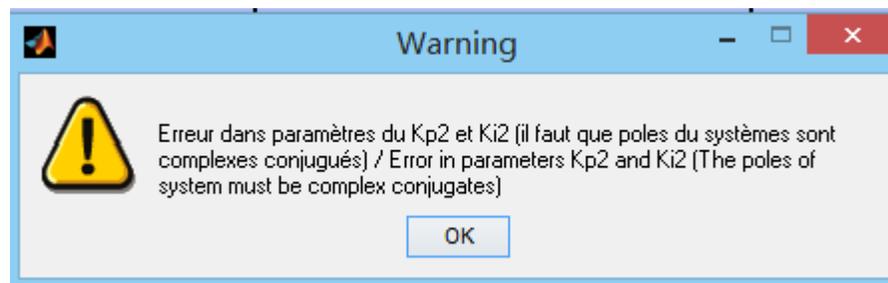


Figure B.4 Message d'erreur No.2.

Pour utiliser cette plate-forme, il faut effectuer les étapes suivantes :

- Définir les paramètres du modèle à paramètres variables, c'est-à-dire que l'utilisateur doit définir les valeurs des paramètres $a_0, b_0; K_p 1, K_i 1; K_p 2, K_i 2$ et les frontières des figures dans les sections [1], [3] et [5].
- Cliquer sur le bouton **Calculer / Calculate** dans la section [7] pour lancer les programmes de calculs et ensuite cliquer sur le bouton **Figures** dans la même section pour tracer les familles de

trajectoires en forme de col et en forme de foyer dans deux figures respectivement. Ces deux figures vont apparaître dans les sections [4] et [6].

- Définir les paramètres de la commande par mode glissant dans la section [8].
- Lorsque toutes les étapes ci-dessus sont accomplies, cliquer sur le bouton, **Calculer / Calculate** dans la section [8] pour faire les calculs et ensuite, cliquer sur le bouton **Figures** dans la même section pour tracer les trajectoires et les réponses temporelles des commandes par modes glissants.

Un exemple de la démonstration de l'application de la commande par mode glissant est présenté à la figure B.5 :

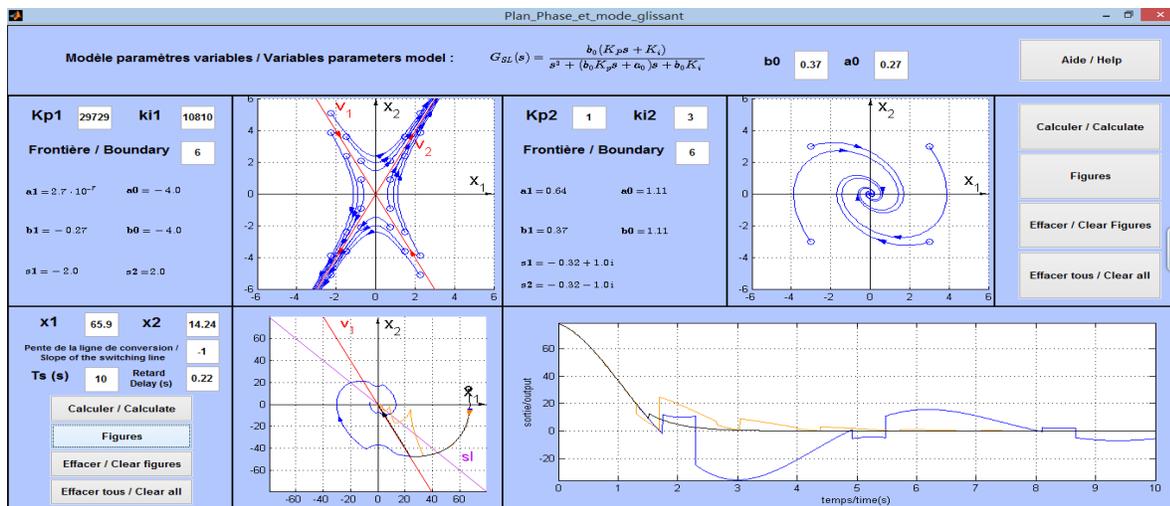


Figure B.5 Visualisation de l'application des commandes par modes glissants.

Les programmes de calculs pour faire l'application de la commande par mode glissant sont construits à l'aide de **Simulink** (voir Fig. B.6) et **Stateflow** (voir Figs. B.7 à B.9).

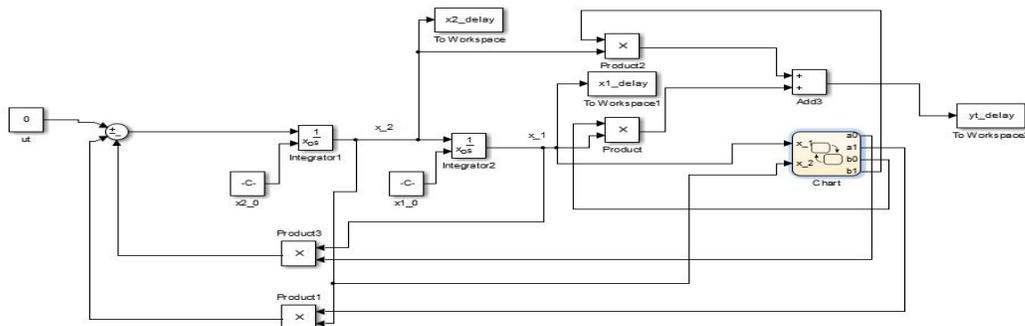


Figure B.6 Schéma de l'application de la commande par mode glissant.

Dans la figure B.6, le module fonctionnel rouge est le module de *Stateflow*. Il est construit pour faire le changement des paramètres du régulateur du système afin de construire la commande par mode glissant. Les schémas de *Stateflow* sont construits pour trois possibilités (voir section [9]) :

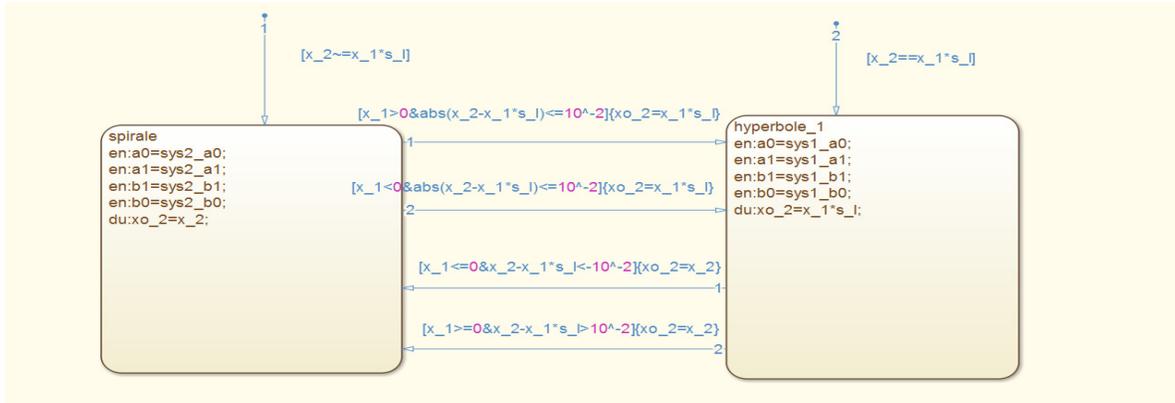


Figure B.7 Schéma de *Stateflow* pour le processus idéal.

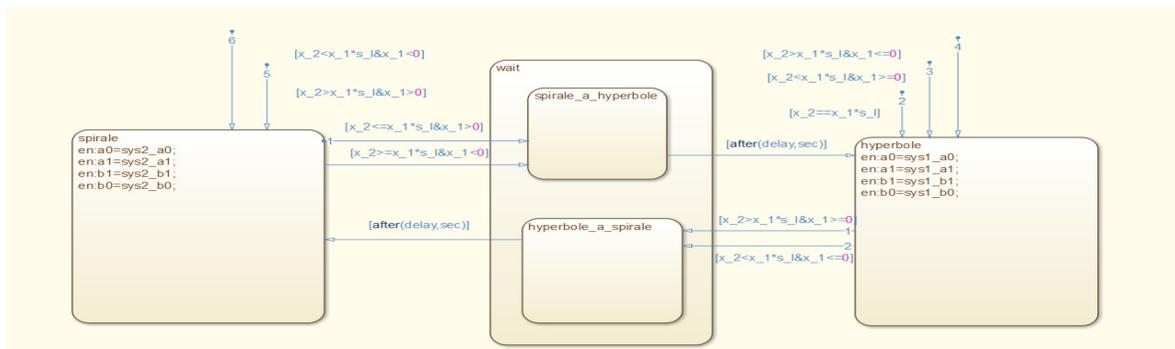


Figure B.8 Schéma de *Stateflow* pour le processus idéal appliqué au système étudié.

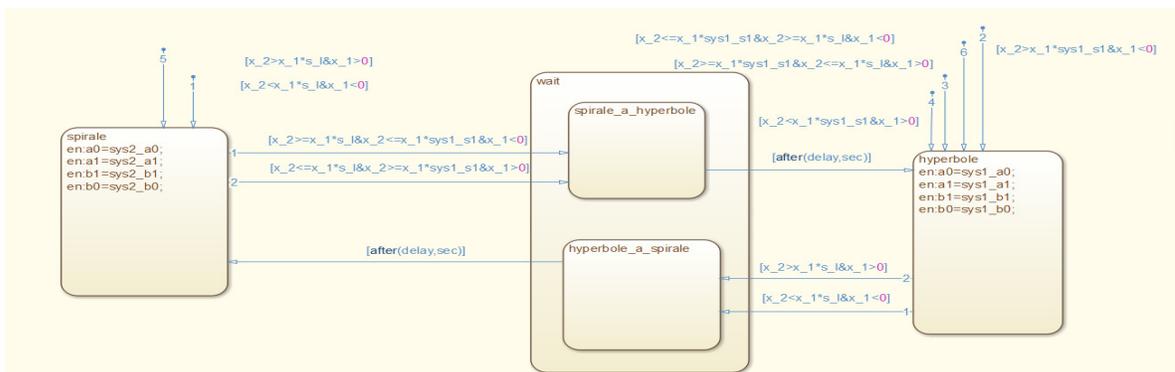


Figure B.9 Schéma de *Stateflow* pour la commande par mode glissant appliquée au système étudié.

Voici une partie des programmations de calculs de l'application de la commande par mode glissant à l'aide de logiciel *Matlab* :

```

%ouvrir des modes du simulink utiliser la sub fonction model_open
model_open
%mettre des parametres pour modes dans simulink % mettre des parametres dans la base
set_param('mode_glissant_ideal','Solver','ode45','MaxStep','1.0000e-04','MinStep','5.0000e-05');
set_param('mode_glissant_delay','Solver','ode45','MaxStep','0.005','MinStep','0.001');
set_param('mode_glissant_delay_2','Solver','ode45','MaxStep','0.005','MinStep','0.001');
set_param('mode_glissant_ideal/ut','Value','0');set_param('mode_glissant_delay/ut','Value','0');
set_param('mode_glissant_delay_2/ut','Value','0');
    %x1_0,x2_0
set_param('mode_glissant_ideal/x1_0','Value','conditions_initiales(1,1)');
set_param('mode_glissant_ideal/x2_0','Value','conditions_initiales(1,2)');
set_param('mode_glissant_delay/x1_0','Value','conditions_initiales(1,1)');
set_param('mode_glissant_delay/x2_0','Value','conditions_initiales(1,2)');
set_param('mode_glissant_delay_2/x1_0','Value','conditions_initiales(1,1)');
set_param('mode_glissant_delay_2/x2_0','Value','conditions_initiales(1,2)');
% mettre parametres des systeme 1 et 2 a workspace
    assignin('base','sys1_a1',sys1_a1);assignin('base','sys1_a0',sys1_a0);assignin('base','sys1_b1',sys1_b1);
    assignin('base','sys1_b0',sys1_b0); assignin('base','sys2_a1',sys2_a1); assignin('base','sys2_a0',sys2_a0);
    assignin('base','sys2_b1',sys2_b1);assignin('base','sys2_b0',sys2_b0);
%simulation des modes dans simulink (mode_glissant_delay,mode_glissant_ideal) %simulation mode glissant
delay pour SL=sys1_s1;
s_l=sys1_s1;assignin('base','s_l',s_l);
assignin('base','sys1_s1',sys1_s1);delay=ci_delay;assignin('base','delay',delay);
%simulation mode
    sim('mode_glissant_delay',[0 conditions_initiales(1,4)]);
    x1_reel=x1_delay;x2_reel=x2_delay;yt_reel=yt_delay
% mettre a workspace
    assignin('base','x1_reel',x1_reel);assignin('base','x2_reel',x2_reel);assignin('base','yt_reel',yt_reel);
%simulation mode glissant ideal pour SL=sys1_s1
    s_l=sys1_s1; assignin('base','s_l',s_l);
%simulation mode
    sim('mode_glissant_ideal',[0 conditions_initiales(1,4)]);
    yt_ideal=[yt_ideal.time,yt_ideal.signals.values];size_yt_ideal=size(yt_ideal);
%effacher point singulier
    n_yt_ideal=size_yt_ideal(1,1);j1=2;
    while j1<n_yt_ideal;
        if abs(yt_ideal(j1,2))~abs(1/2*(yt_ideal(j1-1,2)+yt_ideal(j1+1,2)));
            yt_ideal(j1,2)=1/2*(yt_ideal(j1-1,2)+yt_ideal(j1+1,2));
        else
            end
        j1=j1+1;
    end
% mettre a workspace
    assignin('base','x1_ideal',x1_ideal);assignin('base','x2_ideal',x2_ideal);assignin('base','yt_ideal',yt_ideal);
%simulation mode glissant delay pour SL=sl
    s_l=ci_SL;
    assignin('base','s_l',s_l);assignin('base','sys1_s1',sys1_s1);delay=ci_delay;assignin('base','delay',delay);
%simulation mode %sim('mode_glissant_ocillation',[0 conditions_initiales(1,4)]);
    sim('mode_glissant_delay_2',[0 conditions_initiales(1,4)]);
    x1_ocillation=x1_delay;x2_ocillation=x2_delay;yt_ocillation=yt_delay;
% mettre a workspace
    assignin('base','x1_ocillation',x1_ocillation);assignin('base','x2_ocillation',x2_ocillation);
    assignin('base','yt_ocillation',yt_ocillation);

```

Annexe C : Plate-forme des calculs des transformations géométriques

Cette plate-forme est le deuxième niveau de la plate-forme informatique où on construit cette plate-forme pour transformer le plan de phase au plan d'état afin de trouver la base de la commande glissante. Dans ce cas, sur cette plate-forme informatique, deux transformations géométriques sont calculées et présentées :

1. La transformation du plan de phase au plan d'état et ensuite la démonstration du plan de phase et du plan d'état dans l'espace.
2. La transformation des trajectoires complexes conjuguées à la sphère de *Riemann*.

La plate-forme des calculs des transformations géométriques est construite dans le logiciel *Matlab*. Elle se divise en 10 sections comme la montre dans la figure ci-dessous :

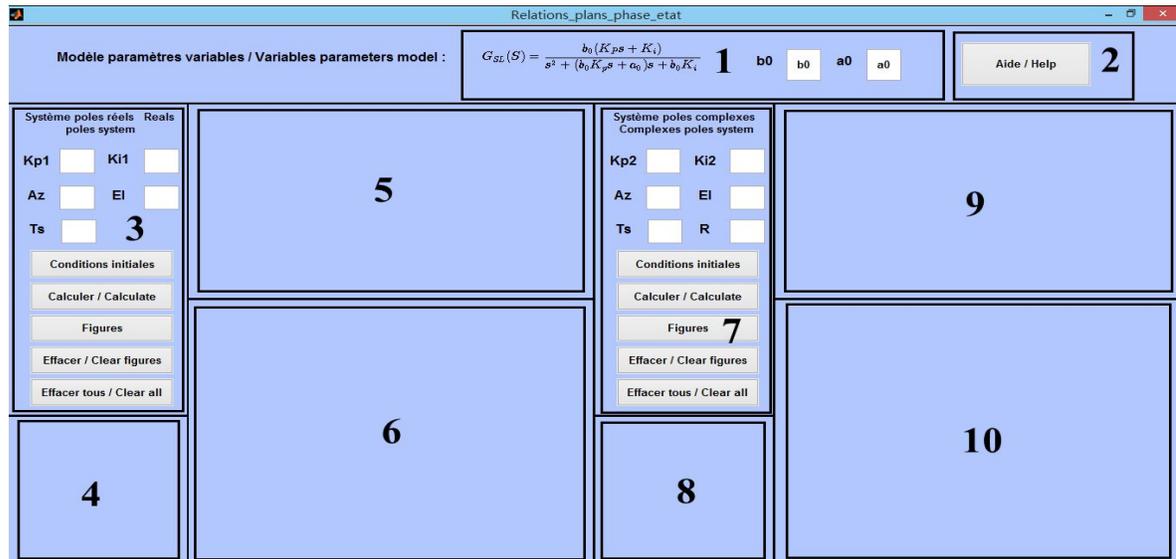


Figure C.1 Interface graphique de la plate-forme des calculs des transformations géométriques.

Description :

- Section [1] : Cette section sert à spécifier les paramètres du modèle à paramètres variables : a_0, b_0 .
- Section [2] : Cette section sert à ouvrir une fenêtre qui présente les informations d'aide de cette plate-forme. La fenêtre d'aide est présentée dans la figure C.2.

- Section [3] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i , les conditions initiales du plan de phase, les paramètres AZ, EL (Spécification de point de vue de la figure dans la section [6]) et le temps de la simulation Ts pour lancer les programmes de calculs afin de tracer les trajectoires dans les figures. Ces figures vont apparaître dans les sections [5] et [6]. Il faut que les pôles du système trouvés par les définitions des paramètres K_p, K_i soient réels. Si l'utilisateur essaye de définir les valeurs des paramètres K_p, K_i pour un système qui ont les pôles complexes conjugués, il reçoit le message d'erreur présentée dans la figure C.3.
- Section [4] : Cette section est dédiée à la visualisation des valeurs des paramètres et à la visualisation des pôles du système. Ce système est défini par les paramètres dans la section [3].
- Section [5] : Cette section est dédiée à la visualisation de la famille de trajectoires en forme de col dans le plan de phase et dans le plan d'état.
- Section [6] : Cette section est dédiée à la visualisation du plan d'état et du plan de phase dans l'espace.
- Section [7] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i , les conditions initiales du plan de phase, les paramètres AZ, EL (Spécification de point de vue de la figure dans la section [10]), le temps de la simulation Ts et le rayon de la sphère **Riemann** R pour lancer les programmes de calculs afin de tracer les trajectoires dans les figures. Ces figures vont apparaître dans les sections [9] et [10]. Il faut que les pôles du système trouvés par les définitions des paramètres K_p, K_i soient complexes conjugués. Si l'utilisateur essaye de définir les valeurs des paramètres K_p, K_i pour un système qui ont les pôles réels, il reçoit le message d'erreur présentée dans la figure C.4.
- Section [8] : Cette section est dédiée à la visualisation des valeurs des paramètres et à la visualisation des pôles du système. Ce système est défini par les paramètres dans la section [7].
- Section [9] : Cette section est dédiée à la visualisation de la famille de trajectoires en forme de foyer dans le plan de phase et à la visualisation des trajectoires complexes conjuguées présentées par la partie réelle et la partie imaginaire dans le plan d'état complexe.
- Section [10] : Cette section est dédiée à la visualisation de la transformation géométrique des trajectoires complexes conjuguées à la sphère de **Riemann**.

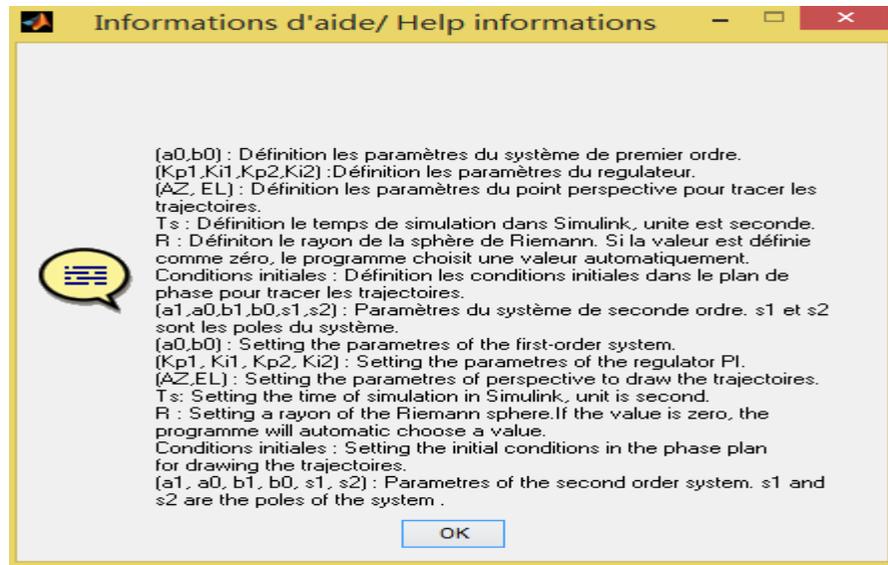


Figure C.2 Fenêtre d'aide.

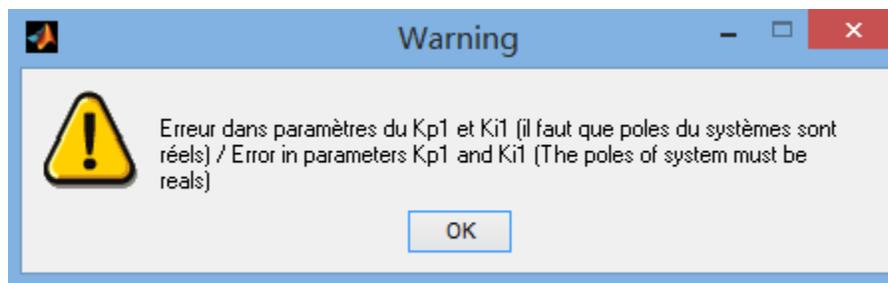


Figure C.3 Message d'erreur No.1.



Figure C.4 Message d'erreur No.2.

Pour utiliser cette plate-forme, il faut effectuer les étapes suivantes :

- Définir les paramètres pour lancer les programmes de calculs afin de trouver la famille de trajectoires en forme de col et faire la démonstration de deux plans dans l'espace. L'utilisateur doit définir les valeurs des paramètres $a_0, b_0; K_p I, K_i I; AZ, EL; Ts$ dans les sections [1] et [3].

- Cliquer sur le bouton **Conditions initiales** dans la section [3] pour définir les conditions initiales dans le plan de phase.
- Cliquer sur le bouton **Calculer / Calculate** dans la section [3] pour lancer les programmes de calculs et ensuite, cliquer sur le bouton **Figures** dans la même section pour tracer les trajectoires dans les figures qui vont apparaître dans les sections [5] et [6].
- Définir les paramètres pour lancer les programmes de calculs afin de trouver la famille de trajectoires en forme de foyer et faire la transformation géométrique des trajectoires complexes conjuguées à la sphère de **Riemann**. L'utilisateur doit définir les valeurs des paramètres K_p, K_i, AZ, EL, Ts, R dans la section [7]. Avant cette étape, si l'utilisateur ne définit pas les valeurs des paramètres a_0, b_0 dans la section [1], toute d'abord, il faut entre les valeurs dans les cases blanches de cette section pour faire la définition.
- Cliquer sur le bouton **Calculer / Calculate** dans la section [7] pour lancer les programmes de calculs et ensuite, cliquer sur le bouton **Figures** dans la même section pour tracer les trajectoires dans les figures qui vont apparaître dans les sections [9] et [10].

Un exemple de la démonstration des transformations géométriques sont présentés à la figure C.5.

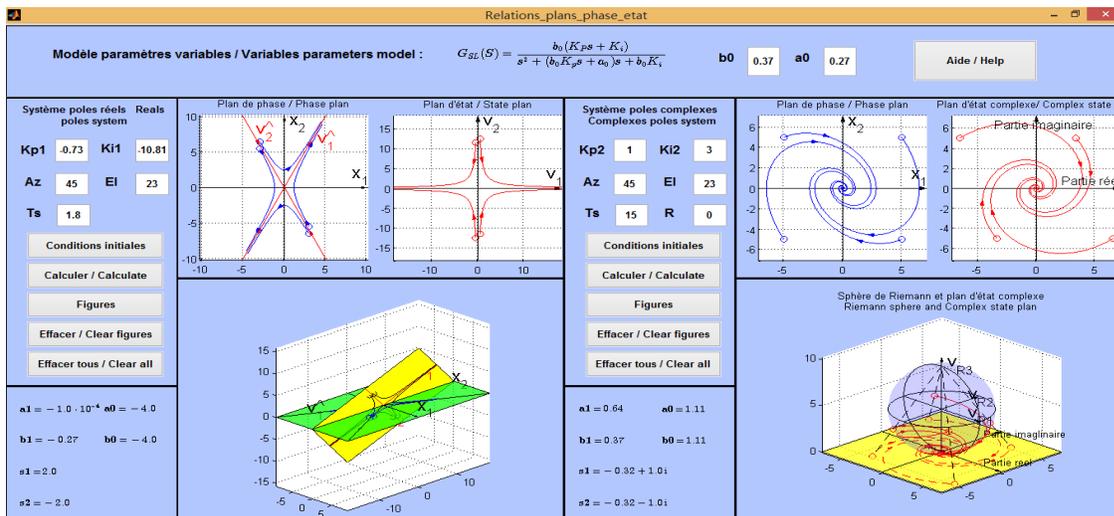


Figure C.5 Visualisation des transformations géométriques.

Voici une partie des programmations de calculs de ces transformations géométriques à l'aide de logiciel **Matlab** :

```

%démonstration du plan de phase et du plan d'état dans l'espace
syms ang4 ang1 ang2 ang3
global v1f v2f x1f x2f
x1f=[sys_reel_vmaxf;0];
x2f=[0;sys_reel_vmaxf];

```

```

v1f=(sys_reel_Mtf^-1)*x1f;
v2f=(sys_reel_Mtf^-1)*x2f;
%%
%%calculer le déplacement des coordonnées d'état.
sys_reel_s1_x1=1;
sys_reel_s1_x2=sys_reel_s1*sys_reel_s1_x1;
sys_reel_s2_x1=1;
sys_reel_s2_x2=sys_reel_s2*sys_reel_s2_x1;
ang_alpha=atan2(sys_reel_s1_x2,sys_reel_s1_x1);
ang_beta=abs(atan2(sys_reel_s2_x2,sys_reel_s2_x1));
angleQ=pi/4+(ang_alpha-ang_beta)/2;
assignin('base','angleQ',angleQ);
%%
%%définir subfonction Solveang
RZN=[cos(ang3),-sin(ang3),0;sin(ang3),cos(ang3),0;0,0,1];
RZS=[cos(ang3),sin(ang3),0;-sin(ang3),cos(ang3),0;0,0,1];
RYN=[cos(ang2),0,sin(ang2);0,1,0;-sin(ang2),0,cos(ang2)];
RYS=[cos(ang2),0,-sin(ang2);0,1,0;sin(ang2),0,cos(ang2)];
RXN=[1,0,0;0,cos(ang1),-sin(ang1);0,sin(ang1),cos(ang1)];
RZN0=[cos(angleQ),-sin(angleQ),0;sin(angleQ),cos(angleQ),0;0,0,1];
RR=RYS*RZS*RXN*RZN*RYN;
RZH=ang4*RR*RZN0;
ZZZ1=RZH*[x1f;0];
ZZZ2=RZH*[x2f;0];
digits(5);
FF1=simple(vpa(ZZZ1(1,1)-v1f(1,1)));
FF2=simple(vpa(ZZZ1(2,1)-v1f(2,1)));
FF3=simple(vpa(ZZZ2(1,1)-v2f(1,1)));
FF4=simple(vpa(ZZZ2(2,1)-v2f(2,1)));
FF1=char(FF1);
FF2=char(FF2);
FF3=char(FF3);
FF4=char(FF4);
FF1=strrep(FF1,'ang4','x(4)');
FF1=strrep(FF1,'ang1','x(1)');
FF1=strrep(FF1,'ang2','x(2)');
FF1=strrep(FF1,'ang3','x(3)');
FF1=sym(FF1);
FF2=strrep(FF2,'ang4','x(4)');
FF2=strrep(FF2,'ang1','x(1)');
FF2=strrep(FF2,'ang2','x(2)');
FF2=strrep(FF2,'ang3','x(3)');
FF2=sym(FF2);
FF3=strrep(FF3,'ang4','x(4)');
FF3=strrep(FF3,'ang1','x(1)');
FF3=strrep(FF3,'ang2','x(2)');
FF3=strrep(FF3,'ang3','x(3)');
FF3=sym(FF3);
FF4=strrep(FF4,'ang4','x(4)');
FF4=strrep(FF4,'ang1','x(1)');
FF4=strrep(FF4,'ang2','x(2)');
FF4=strrep(FF4,'ang3','x(3)');
FF4=sym(FF4);
FFT=[FF1;FF2;FF3;FF4];

```

```

FFT=char(FFT);
FFT=strep(FFT,'matrix','F=');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%écrire dans le document de la subfonction Solveang
fid=fopen('Solveang.m','w+');
fprintf(fid,'function F = Solveang(x)\r\n');
%fprintf(fid,'global x1f x2f v1f v2f;\r\n');
fprintf(fid,FFT);
pause(2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculer les valeurs des paramètres
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculer des angles des rotations;
solve_angle=fsolve('Solveang',[1,1,0,1]); assignin('base','solve_angle',solve_angle);
%définir des matrices des rotations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RZN=[cos(solve_angle(1,3)),sin(solve_angle(1,3)),0;sin(solve_angle(1,3)),cos(solve_angle(1,3)),0;0,0,1];
RZS=[cos(solve_angle(1,3)),sin(solve_angle(1,3)),0;-sin(solve_angle(1,3)),cos(solve_angle(1,3)),0;0,0,1];
RYN=[cos(solve_angle(1,2)),0,sin(solve_angle(1,2));0,1,0;-sin(solve_angle(1,2)),0,cos(solve_angle(1,2))];
RYS=[cos(solve_angle(1,2)),0,-sin(solve_angle(1,2));0,1,0;sin(solve_angle(1,2)),0,cos(solve_angle(1,2))];
RXN=[1,0,0;0,cos(solve_angle(1,1)),sin(solve_angle(1,1));0,sin(solve_angle(1,1)),cos(solve_angle(1,1))];
RZN0=[cos(angleQ),-sin(angleQ),0;sin(angleQ),cos(angleQ),0;0,0,1];
RR=RYS*RZS*RXN*RZN*RYN;
RZH=solve_angle(1,4)*RR*RZN0;
assignin('base','RZH',RZH);
%transformation les trajectoires complexes conjuguées à la sphère de Riemann
%définir v2_reel et v2_imaginaire
for i8=1:size_conditionsX_sys_complexe(1,2);
    sys_complexe_VS0(:,i8)=sys_complexe_Mtf*conditions_initiales_sys_complexe(:,i8);
    V2_reel=real(sys_complexe_VS0(2,i8));
    V2_imaginaire=imag(sys_complexe_VS0(2,i8));
    VS2(1,i8)=V2_reel;
    VS2(2,i8)=V2_imaginaire;
end
assignin('base','sys_complexe_VS0',sys_complexe_VS0);
assignin('base','VS2',VS2);
for i9=1:size_conditionsX_sys_complexe(1,2);
    sys_complexe_v_VC(1,1)=VS2(1,i9);
    sys_complexe_v_VC(1,2)=-1*(VS2(2,i9));
    sys_complexe_v_VS0=[sys_complexe_v_VC(1,1)-1*sys_complexe_v_VC(1,2)];
    sys_complexe_v_VS2=[sys_complexe_v_VC(1,1)-1*sys_complexe_v_VC(1,2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% faire la transformation à la sphère de Riemann
    for i4=2:length(sys_complexe_T0);
        sys_complexe_v_tt=sys_complexe_T0(1,i4);
        sys_complexe_v_VS0(1,1)=exp(sys_complexe_alpha*sys_complexe_v_tt)*((sys_complexe_v_VC(1,1)*cos(
        sys_complexe_beta*sys_complexe_v_tt)-
        sys_complexe_v_VC(1,2)*sin(sys_complexe_beta*sys_complexe_v_tt));
        sys_complexe_v_VS0(1,2)=-
        1*exp(sys_complexe_alpha*sys_complexe_v_tt)*((sys_complexe_v_VC(1,2)*cos(sys_complexe_beta*sys_c
        omplexe_v_tt)+sys_complexe_v_VC(1,1)*sin(sys_complexe_beta*sys_complexe_v_tt));
        sys_complexe_v_VS2=[sys_complexe_v_VS2;sys_complexe_v_VS0];
    end
    size_sys_complexe_v_VS2=size(sys_complexe_v_VS2);
    assignin('base','size_sys_complexe_v_VS2',size_sys_complexe_v_VS2);
    for i5=1:size_sys_complexe_v_VS2;
        VS21(i5,1:i9)=sys_complexe_v_VS2(i5,:);
    end

```

```

    end
end
assignin('base','VS21',VS21);
sizeVS21=size(VS21);
assignin('base','sizeVS21',sizeVS21);
assignin('base','sys_complexe_T0',sys_complexe_T0);

sizeVS21=size(sizeVS21);
assignin('base','sizeVS21',sizeVS21);

if sizeVS21(1,2)==3;
    for j3=1:sizeVS21(1,3);
        VS22=zeros(sizeVS21(1,1),1);
        VS21E(:,j3)=[VS21(:,j3),VS22];
    end
    assignin('base','VS21E',VS21E);
elseif sizeVS21(1,2)==2;
    VS22=zeros(sizeVS21(1,1),1);
    VS21E(:,j3)=[VS21(:,j3),VS22];
    assignin('base','VS21E',VS21E);
end
% %%% définir des limites des frontières dans plan d'état
if sizeVS21(1,2)==2;
    V2maxmin(1,1)=max(VS21(1:end,1));
    V2maxmin(1,2)=max(VS21(1:end,2));
    V2maxmin(1,3)=-1*min(VS21(1:end,1));
    V2maxmin(1,4)=-1*min(VS21(1:end,2));
    assignin('base','V2maxmin',V2maxmin);
elseif sizeVS21(1,2)==3;
    for j3=1:sizeVS21(1,3);
        V2maxmin(1,1,j3)=max(VS21(1:end,1,j3));
        V2maxmin(1,2,j3)=max(VS21(1:end,2,j3));
        V2maxmin(1,3,j3)=-1*min(VS21(1:end,1,j3));
        V2maxmin(1,4,j3)=-1*min(VS21(1:end,2,j3));
    end
    assignin('base','V2maxmin',V2maxmin);
end
end
Vmax=1.1*max(V2maxmin);
Vmaxf=max(Vmax(1,1,1:end));
assignin('base','Vmax',Vmax);
assignin('base','Vmaxf',Vmaxf);

% %%% définir le rayon de la sphère de Riemann (auto)
if sys_complexe_r==0;
    sys_complexe_r=Vmaxf/pi*2;
parametres_initiales_sys_complexe(1,8)=sys_complexe_r;
assignin('base','parametres_initiales_sys_complexe',parametres_initiales_sys_complexe);
else
end
end
% %%% calculer des point dans la sphère de Riemann
if sizeVS21(1,2)==3;
    ES1=zeros(sizeVS21(1,1),sizeVS21(1,2)+1,sizeVS21(1,3));
    sizeES1=size(ES1);
    assignin('base','sizeES1',sizeES1);
    sizeES1=size(sizeES1);
    assignin('base','sizeES1',sizeES1);
end

```

```

for j3=1:sizeES1(1,3);
for j4=1:sizeES1(1,1);
VE1=(4*sys_complexe_r^2*VS21(j4,1,j3))/(VS21(j4,1,j3)^2+VS21(j4,2,j3)^2+4*sys_complexe_r^2);
VE2=(4*sys_complexe_r^2*VS21(j4,2,j3))/(VS21(j4,1,j3)^2+VS21(j4,2,j3)^2+4*sys_complexe_r^2);

VE3=(2*sys_complexe_r*(VS21(j4,1,j3)^2+VS21(j4,2,j3)^2))/(VS21(j4,1,j3)^2+VS21(j4,2,j3)^2+4*sys
_complexe_r^2);
ES1(j4,:j3)=[VE1,VE2,VE3];
end
end
assignin('base','ES1',ES1);
elseif sizesizeVS21(1,2)==2;
ES1=zeros(sizeVS21(1,1),sizeVS21(1,2)+1);
sizeES1=size(ES1);
assignin('base','sizeES1',sizeES1);
sizesizeES1=size(sizeES1);
assignin('base','sizesizeES1',sizesizeES1);
for j4=1:sizeES1(1,1);
VE1=(4*sys_complexe_r^2*VS21(j4,1))/(VS21(j4,1)^2+VS21(j4,2)^2+4*sys_complexe_r^2);
VE2=(4*sys_complexe_r^2*VS21(j4,2))/(VS21(j4,1)^2+VS21(j4,2)^2+4*sys_complexe_r^2);

VE3=(2*sys_complexe_r*(VS21(j4,1)^2+VS21(j4,2)^2))/(VS21(j4,1)^2+VS21(j4,2)^2+4*sys_complexe_r^
2);
ES1(j4,:)= [VE1,VE2,VE3];
end
assignin('base','ES1',ES1);
end
%%%%%% définir des limites des frontières de la sphère
if sizesizeES1(1,2)==2;
ESmaxmin(1,1)=max(ES1(1:end,1));
ESmaxmin(1,2)=max(ES1(1:end,2));
ESmaxmin(1,3)=max(ES1(1:end,3));
ESmaxmin(1,4)=-1*min(ES1(1:end,1));
ESmaxmin(1,5)=-1*min(ES1(1:end,2));
ESmaxmin(1,6)=-1*min(ES1(1:end,3));

elseif sizesizeES1(1,2)==3;
for j3=1:sizeES1(1,3);
ESmaxmin(1,1,j3)=max(ES1(1:end,1,j3));
ESmaxmin(1,2,j3)=max(ES1(1:end,2,j3));
ESmaxmin(1,3,j3)=max(ES1(1:end,3,j3));
ESmaxmin(1,4,j3)=-1*min(ES1(1:end,1,j3));
ESmaxmin(1,5,j3)=-1*min(ES1(1:end,2,j3));
ESmaxmin(1,6,j3)=-1*min(ES1(1:end,3,j3));
end
end
esmax=1.1*max(ESmaxmin);
esmaxf=max(esmax(1,1,1:end));
esmaxf=max(esmaxf,2*1.1*sys_complexe_r);
assignin('base','esmaxf',esmaxf);

```

Annexe D : Plate-forme des calculs des commandes glissantes

Cette plate-forme est le troisième niveau de la plate-forme informatique où on construit cette plate-forme pour obtenir des informations et pour faire la visualisation d'application de la commande glissante.

La plate-forme des calculs des commandes glissantes est construite dans le logiciel *Matlab*. Elle se divise en 9 sections comme la montre dans la figure ci-dessous :

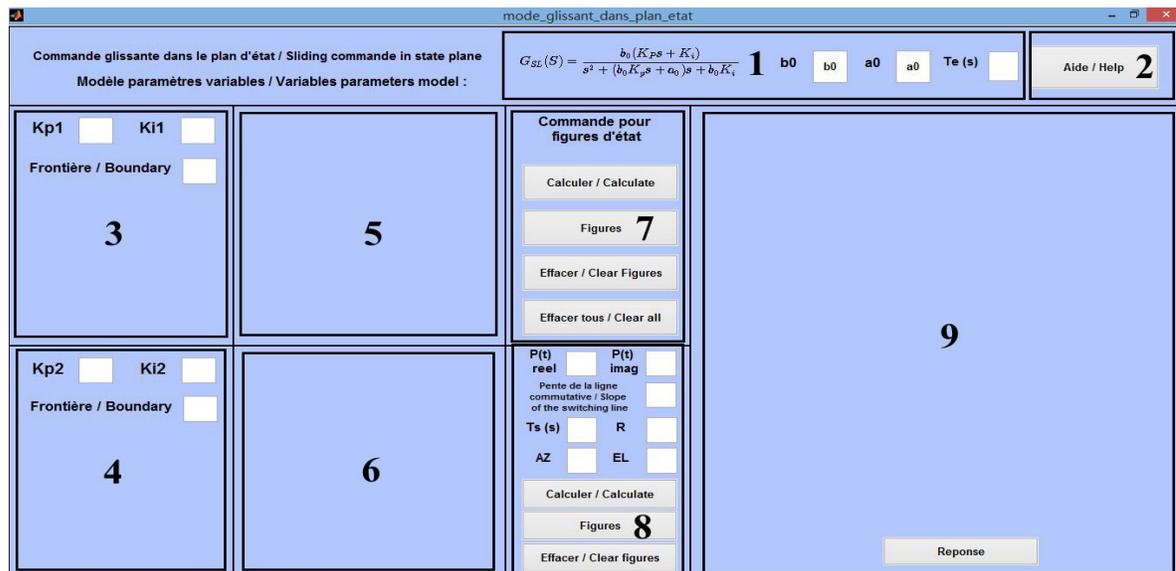


Figure D.1 Interface graphique de la plate-forme des calculs des commandes glissantes.

Description :

- Section [1] : Cette section sert à spécifier les paramètres du modèle à paramètres variables, a_0, b_0 et à spécifier la période d'échantillonnage $Te(s)$.
- Section [2] : Cette section sert à ouvrir une fenêtre qui présente les informations d'aide de cette plate-forme. La fenêtre d'aide est présentée dans la figure D.2.
- Section [3] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i et à définir la frontière de la figure pour tracer les trajectoires dans la section [5]. Il faut que les pôles du système trouvés par les définitions des paramètres K_p, K_i soient réels. Si l'utilisateur essaye de définir les valeurs des paramètres K_p, K_i pour un système qui ont les pôles complexes conjugués, il reçoit le message d'erreur présentée dans la figure D.3.

- Section [4] : Cette section sert à spécifier les paramètres du correcteur K_p, K_i et à définir la frontière de la figure pour tracer les trajectoires dans la section [6]. Il faut que les pôles du système trouvés par les définitions des paramètres K_p, K_i soient complexes conjugués. Si l'utilisateur essaye de définir les valeurs des paramètres K_p, K_i pour un système qui ont les pôles réels, il reçoit le message d'erreur présentée dans la figure D.4
- Section [5] : Cette section est dédiée à la visualisation de la famille de trajectoires en forme de col dans le plan d'état.
- Section [6] : Cette section est dédiée à la visualisation des trajectoires complexes conjuguées dans le plan d'état complexe présentées par la partie réelle et la partie imaginaire.
- Section [7] : Cette section est consacrée aux calculs pour trouver les trajectoires qui sont présentées dans les sections [5] et [6].
- Section [8] : Cette section sert à spécifier les paramètres de la démonstration de l'application de la commande glissante : le point initial présenté par les parties réelle et imaginaire $(P_{Reel}(0), P_{Imaginaire}(0))$. Il est utilisé pour trouver les conditions initiales sur la sphère de **Riemann**. La ligne commutative, le temps de simulation $Ts(s)$, le rayon de la sphère de **Riemann** R et les paramètres AZ, EL (Spécification de point de vue de la figure dans la section [9]).
- Section [9] : Cette section est dédiée à la visualisation de la trajectoire de la commande glissante.

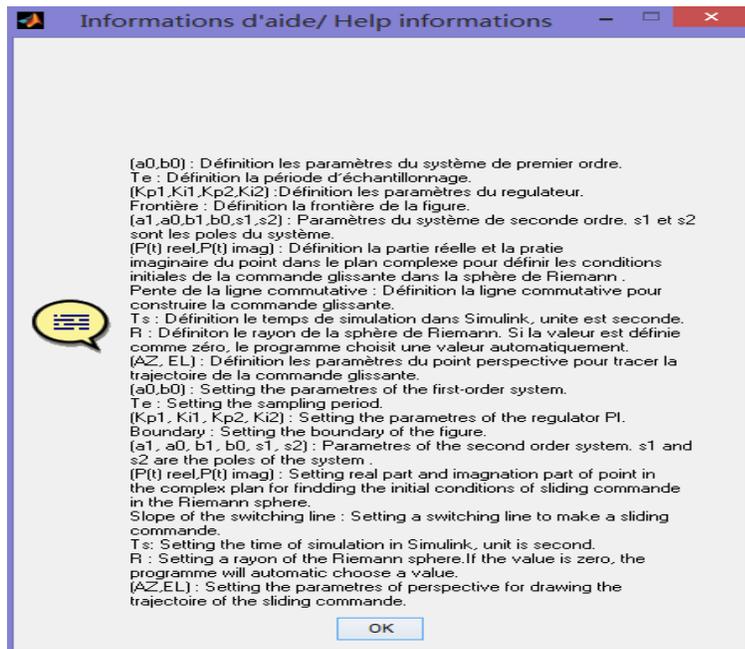


Figure D.2 Fenêtre d'aide.



Figure D.3 Message d'erreur No.1.



Figure D.4 Message d'erreur No.2.

Pour utiliser cette plate-forme, il faut effectuer les étapes suivantes :

- Définir les paramètres du modèle à paramètres variables, c'est-à-dire que l'utilisateur doit définir les valeurs des paramètres $a_0, b_0; K_p1, K_i1; K_p2, K_i2$ et les frontières des figures dans les sections [1], [3] et [4].
- Cliquer sur le bouton **Calculer / Calculate** dans la section [7] pour lancer les programmes de calculs afin d'obtenir les trajectoires et ensuite, cliquer sur le bouton **Figures** dans la même section pour tracer les trajectoires dans deux figures qui vont apparaitre dans les sections [5] et [6].
- Définir les paramètres de la commande glissante dans les sections [1] et [8].
- Lorsque toutes les étapes ci-dessus sont accomplies, cliquer sur le bouton, **Calculer / Calculate** dans la section [8] pour faire les calculs et ensuite, cliquer sur le bouton **Figures** dans la même section pour tracer la trajectoire de la commande glissante. Cliquer sur le bouton **Reponse** dans la section [9] pour ouvrir la figure qui présente la réponse temporelle de la commande glissante.

Un exemple de la démonstration de l'application de la commande glissante est présenté à la figure D.5 :

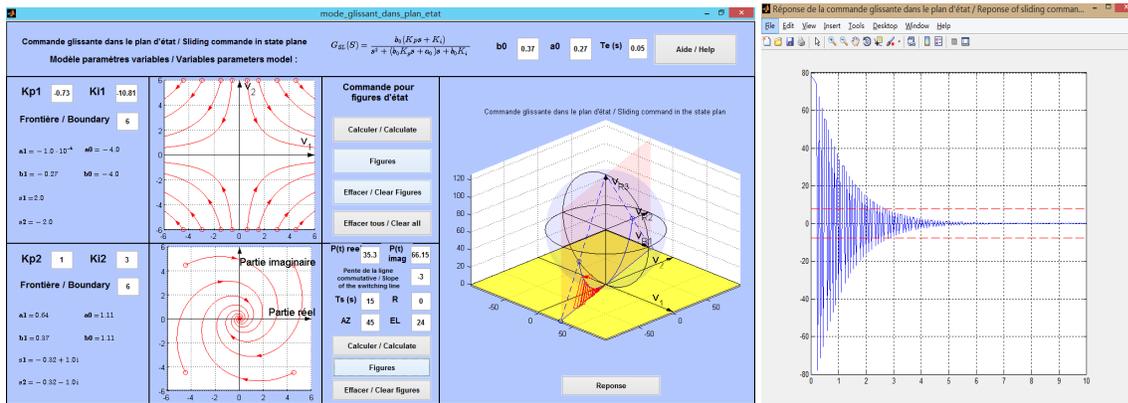


Figure D.5 Visualisation de l'application de la commande glissante.

Voici une partie des programmations de l'application de la commande glissante à l'aide de **Matlab** :

```

%%définition les paramètres de la commande glissante
sys_complexe_alpha=evalin('base','sys_complexe_alpha');
sys_complexe_beta=evalin('base','sys_complexe_beta');
sys_complexe_gama=evalin('base','sys_complexe_gama');
sys_complexe_sigma=evalin('base','sys_complexe_sigma');
sys_reel_s1=evalin('base','sys_reel_s1');
sys_reel_s2=evalin('base','sys_reel_s2');
sys_reel_c1=evalin('base','sys_reel_c1');
sys_reel_c2=evalin('base','sys_reel_c2');
v2_reel=parametres_mode_glissant_etat(1,1);
v2_imag=parametres_mode_glissant_etat(1,2);
v20_reel=parametres_mode_glissant_etat(1,1);
v20_imag=parametres_mode_glissant_etat(1,2);
te=parametres_mode_glissant_etat(1,8);
ts=parametres_mode_glissant_etat(1,4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% définition le temps de simulation, la période d'échantillonnage et les
conditions initiales
t00=0:te:ts;
v_surface=[parametres_mode_glissant_etat(1,1),parametres_mode_glissant_etat(1,2)];

vr1_espace=4*sys_mode_glissant_etat_r^2*v2_reel/(v2_reel^2+v2_imag^2+4*sys_mode_glissant_etat_r^2);
vr2_espace=4*sys_mode_glissant_etat_r^2*v2_imag/(v2_reel^2+v2_imag^2+4*sys_mode_glissant_etat_r^2);

vr3_espace=2*sys_mode_glissant_etat_r*(v2_reel^2+v2_imag^2)/(v2_reel^2+v2_imag^2+4*sys_mode_glissant_etat_r^2);
yt=2*(sys_complexe_sigma*v2_reel-sys_complexe_gama*v2_imag);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% simulation de l'application de la commande glissante
for i3=2:length(t00);
    if v20_reel>=0&v20_imag>sys_mode_glissant_etat_s1*v20_reel;
        tt=te;
        vr1=4*sys_mode_glissant_etat_r^2*v20_reel/(v20_reel^2+v20_imag^2+4*sys_mode_glissant_etat_r^2);
vr2=4*sys_mode_glissant_etat_r^2*v20_imag/(v20_reel^2+v20_imag^2+4*sys_mode_glissant_etat_r^2);

```



```

vr3=2*sys_mode_glissant_etat_r*(v20_reel^2+v20_imag^2)/(v20_reel^2+v20_imag^2+4*sys_mode_glissant_
_ _etat_r^2);
    vr1_espace=[vr1_espace;vr1];
    vr2_espace=[vr2_espace;vr2];
    vr3_espace=[vr3_espace;vr3];
v22_reel=exp(sys_complexe_alpha*tt)*(v20_reel*cos(sys_complexe_beta*tt)+v20_imag*sin(sys_complexe_
beta*tt));
v22_imag=1*exp(sys_complexe_alpha*tt)*(1*v20_imag*cos(sys_complexe_beta*tt)+v20_reel*sin(sys_com
plexe_beta*tt));
v_surface=[v_surface;v22_reel,v22_imag];
vr1=4*sys_mode_glissant_etat_r^2*v22_reel/(v22_reel^2+v22_imag^2+4*sys_mode_glissant_etat_r^2);
vr2=4*sys_mode_glissant_etat_r^2*v22_imag/(v22_reel^2+v22_imag^2+4*sys_mode_glissant_etat_r^2);
vr3=2*sys_mode_glissant_etat_r*(v22_reel^2+v22_imag^2)/(v22_reel^2+v22_imag^2+4*sys_mode_glissant_
_ _etat_r^2);
    vr1_espace=[vr1_espace;vr1];
    vr2_espace=[vr2_espace;vr2];
    vr3_espace=[vr3_espace;vr3];
    yt_sortie=2*(sys_complexe_sigma*v22_reel-sys_complexe_gama*v22_imag);
    yt=[yt;yt_sortie];
    v20_reel=v22_reel;
    v20_imag=v22_imag;
elseif v20_reel<=0&v20_imag<sys_mode_glissant_etat_sl*v20_reel;
    tt=te;
    vr1=4*sys_mode_glissant_etat_r^2*v20_reel/(v20_reel^2+v20_imag^2+4*sys_mode_glissant_etat_r^2);
    vr2=4*sys_mode_glissant_etat_r^2*v20_imag/(v20_reel^2+v20_imag^2+4*sys_mode_glissant_etat_r^2);
    vr3=2*sys_mode_glissant_etat_r*(v20_reel^2+v20_imag^2)/(v20_reel^2+v20_imag^2+4*sys_mode_glis
sant_etat_r^2);
    vr1_espace=[vr1_espace;vr1];
    vr2_espace=[vr2_espace;vr2];
    vr3_espace=[vr3_espace;vr3];
    v22_reel=exp(sys_complexe_alpha*tt)*(v20_reel*cos(sys_complexe_beta*tt)+v20_imag*sin(sys_co
mplexe_beta*tt));
    v22_imag=1*exp(sys_complexe_alpha*tt)*(1*v20_imag*cos(sys_complexe_beta*tt)+v20_reel*sin(
sys_complexe_beta*tt));
    v_surface=[v_surface;v22_reel,v22_imag];
vr1=4*sys_mode_glissant_etat_r^2*v22_reel/(v22_reel^2+v22_imag^2+4*sys_mode_glissant_etat_r^2);
vr2=4*sys_mode_glissant_etat_r^2*v22_imag/(v22_reel^2+v22_imag^2+4*sys_mode_glissant_etat_r^2);
vr3=2*sys_mode_glissant_etat_r*(v22_reel^2+v22_imag^2)/(v22_reel^2+v22_imag^2+4*sys_mode_glis
sant_etat_r^2);
    vr1_espace=[vr1_espace;vr1];
    vr2_espace=[vr2_espace;vr2];
    vr3_espace=[vr3_espace;vr3];
    yt_sortie=2*(sys_complexe_sigma*v22_reel-sys_complexe_gama*v22_imag);
    yt=[yt;yt_sortie];
    v20_reel=v22_reel;
    v20_imag=v22_imag;
elseif v20_reel<0&v20_imag>=sys_mode_glissant_etat_sl*v20_reel;
    tt=te
    vr1=v20_reel;
    vr2=v20_imag;
    vr3=0;
    vr1_espace=[vr1_espace;vr1];
    vr2_espace=[vr2_espace;vr2];
    vr3_espace=[vr3_espace;vr3];
    v22_reel=exp(sys_reel_sl*tt)*v20_reel;
    v22_imag=(v22_reel/v20_reel)^(sys_reel_s2/sys_reel_s1)*v20_imag;

```

```

v_surface=[v_surface;v22_reel,v22_imag];
vr1=v22_reel;
vr2=v22_imag;
vr3=0;
vr1_espace=[vr1_espace;vr1];
vr2_espace=[vr2_espace;vr2];
vr3_espace=[vr3_espace;vr3];
yt_sortie=sys_reel_c1*v22_reel+sys_reel_c2*v22_imag;
yt=[yt;yt_sortie];
v20_reel=v22_reel;
v20_imag=v22_imag;
elseif v20_reel>0&v20_imag<=sys_mode_glissant_etat_sl*v20_reel;
tt=te
vr1=v20_reel;
vr2=v20_imag;
vr3=0;
vr1_espace=[vr1_espace;vr1];
vr2_espace=[vr2_espace;vr2];
vr3_espace=[vr3_espace;vr3];
v22_reel=exp(sys_reel_s1*tt)*v20_reel;
v22_imag=(v22_reel/v20_reel)^(sys_reel_s2/sys_reel_s1)*v20_imag;
v_surface=[v_surface;v22_reel,v22_imag];
vr1=v22_reel;
vr2=v22_imag;
vr3=0;
vr1_espace=[vr1_espace;vr1];
vr2_espace=[vr2_espace;vr2];
vr3_espace=[vr3_espace;vr3];
yt_sortie=sys_reel_c1*v22_reel+sys_reel_c2*v22_imag;
yt=[yt;yt_sortie];
v20_reel=v22_reel;
v20_imag=v22_imag;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% déplacement les résultats au work space dans le matlab
assignin('base','v22_reel',v22_reel);
assignin('base','v22_imag',v22_imag);
assignin('base','v_surface',v_surface);
assignin('base','vr1_espace',vr1_espace);
assignin('base','vr2_espace',vr2_espace);
assignin('base','vr3_espace',vr3_espace);
assignin('base','yt',yt);
assignin('base','t00',t00);

```