Robustification of Nonlinear Model Predictive Control - Application to sustainable development processes.

Seif Eddine Benattia

NNT : 2016SACLC066

THESE DE DOCTORAT
DE
L'UNIVERSITE PARIS-SACLAY
PREPAREE A
CENTRALESUPELEC

ECOLE DOCTORALE N° 580
Science et technologies de l'information et de la communication

Spécialité de doctorat : Automatique

Par

M. Seif Eddine Benattia

Robustification de la commande prédictive non linéaire - Applications à des procédés pour le développement durable

Thèse présentée et soutenue à Gif-sur-Yvette, le 21 Septembre 2016 :

Composition du Jury :

Mme Estelle Courtial	Maître de Conférences, Université d'Orléans	Examinatrice
M. Didier Dumur	Professeur, CentraleSupélec/L2S	Directeur de thèse
M. Hugues Mounier	Professeur, Université Paris Sud/L2S	Président
M. Mohammed M'Saad	Professeur, ENSICAEN	Rapporteur
Mme Sihem Tebbani	Professeure associée, CentraleSupélec/L2S	Co-encadrante
M. Alain Vande Wouwer	Professeur, Université de Mons	Rapporteur

Dédicaces

Je dédie ce modeste travail :

A mes chers parents Yasmina et Djamel Eddine. Loin de vous, votre sacrifice et votre amour m'ont toujours donné de la force pour prospérer dans la vie. J'ose espérer que ma mère verra à travers cette thèse de doctorat une sorte de concrétisation de tous les efforts qu'elle a fournis pour m'éduquer tout au long de sa vie.

A tonton Mostefa paix à son âme. J'adresse une pensée toute particulière à celui qui restera pour moi une inspiration pour donner le meilleur de moi même.

A ma deuxième mère tata Farah qui m'a accueilli à bras ouverts. En témoignage de l'attachement et de l'affection que je te porte.

A ma tendre femme Kaouter pour son éternel soutien, sa patience et surtout sa précieuse aide chaque fois que nécessaire.

A mes deux frères Abdelhamid et Mohamed Amine ainsi qu'à leurs femmes et ma nièce Manessa.

A ma belle famille et spécialement Djaoued, Ilyes, Asma, Alya et Mami.

Remerciements

Le travail présenté dans ce mémoire a été mené à CentraleSupélec/Laboratoire des Signaux et Systèmes (L2S).

Je tiens à exprimer ma profonde gratitude au Professeur Didier Dumur pour m'avoir accueilli dans son équipe au sein du L2S. Sa gentillesse, sa disponibilité sur le plan professionnel et humain, ses compétences scientifiques m'ont permis d'effectuer ce travail dans les meilleures conditions. Je remercie également Madame Sihem Tebbani qui a co-encadré cette thèse, pour tous les conseils qu'elle m'a prodigués, la confiance et le suivi régulier qu'elle m'a accordés qui ont largement contribué à rendre ces années de thèse très agréables.

Je remercie le Professeur Hugues Mounier pour avoir présidé ce jury. Je suis également honoré que les professeurs Mohammed M'Saad et Alain Vande Wouwer aient accepté d'être mes rapporteurs. Malgré leurs emplois du temps chargés, ils ont pris le temps de juger mon travail. Je les en remercie vivement. Je tiens également à adresser mes sincères remerciements à Madame Estelle Courtial qui a accepté de faire partie du jury.

Je remercie également tout le personnel du département Automatique de CentraleSupélec et toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail. Enfin, je remercie tous mes amis et collègues qui ont été présents dans les bons moments comme dans les plus difficiles et en particulier : Sofiane, Djawad, Djamal, Zaki, Salim, Imad, Tahar, Adlene, Fethi, Mircea, Tri et Idir.

Résumé

Les dernières années ont permis des développements très rapides, tant au niveau de l'élaboration que de l'application, d'algorithmes de commande prédictive non linéaire (CPNL), avec une gamme relativement large de réalisations industrielles. Un des obstacles les plus significatifs rencontré lors du développement de cette commande est lié aux incertitudes sur le modèle du système.

Dans ce contexte, l'objectif principal de cette thèse est la conception de lois de commande prédictives non linéaires robustes vis-à-vis des incertitudes sur le modèle. Classiquement, cette synthèse peut s'obtenir via la résolution d'un problème d'optimisation min-max. L'idée est alors de minimiser l'erreur de suivi de la trajectoire optimale pour la pire réalisation d'incertitudes possible. Cependant, cette formulation de la commande prédictive robuste induit une complexité qui peut être élevée ainsi qu'une charge de calcul importante, notamment dans le cas de systèmes multivariables, avec un nombre de paramètres incertains élevé. Pour y remédier, la principale approche proposée dans ces travaux consiste à simplifier le problème d'optimisation min-max, via l'analyse de sensibilité du modèle vis-à-vis de ses paramètres afin d'en réduire le temps de calcul.

Dans un premier temps, le critère est linéarisé autour des valeurs nominales des paramètres du modèle. Les variables d'optimisation sont soit les commandes du système soit l'incrément de commande sur l'horizon temporel. Le problème d'optimisation initial est alors transformé soit en un problème convexe, soit en un problème de minimisation unidimensionnel, en fonction des contraintes imposées sur les états et les commandes. Une analyse de la stabilité du système en boucle fermée est également proposée.

En dernier lieu, une structure de commande hiérarchisée combinant la commande prédictive robuste linéarisée et une commande par mode glissant intégral est développée afin d'éliminer toute erreur statique en suivi de trajectoire de référence. L'ensemble des stratégies proposées est appliqué à deux cas d'études de commande de bioréacteurs de culture de microorganismes.

Executive Summary

The last few years have led to very rapid developments, both in the formulation and the application of Nonlinear Model Predictive Control (NMPC) algorithms, with a relatively wide range of industrial achievements. One of the most significant challenges encountered during the development of this control law is due to uncertainties in the model of the system.

In this context, the thesis addresses the design of NMPC control laws robust towards model uncertainties. Usually, the above design can be achieved through solving a min-max optimization problem. In this case, the idea is to minimize the tracking error for the worst possible uncertainty realization. However, this robust approach tends to become too complex to be solved numerically online, especially in the case of multivariable systems with a large number of uncertain parameters. To address this shortfall, the main proposed approach consists in simplifying the min-max optimization problem through a sensitivity analysis of the model with respect to its parameters, in order to reduce the calculation time.

First, the criterion is linearized around the model parameters nominal values. The optimization variables are either the system control inputs or the control increments over the prediction horizon. The initial optimization problem is then converted either into a convex optimization problem, or a one-dimensional minimization problem, depending on the nature of the constraints on the states and commands. The stability analysis of the closed-loop system is also addressed.

Finally, a hierarchical control strategy is developed, that combines a robust model predictive control law with an integral sliding mode controller, in order to cancel any tracking error. The proposed approaches are applied through two case studies to the control of microorganisms culture in bioreactors.

Contents

List of Figures

LIST OF FIGURES

List of Tables

Acronyms

asNMPC	advanced step Nonlinear Model Predictive Control
BMI	Bilinear Matrix Inequality
CLRMPC	Constrained Linearized Robust Model Predictive Controller
CSTR	Continuous Stirred Tank Reactor
CVP	Control Vector Parametrization
DMC	Dynamic Matrix Control
DMS	Direct Multiple Shooting
DPC	Distributed Predictive Control
DSS	Direct Simple Shooting
D-RTO	Dynamic Real Time Optimization
EHAC	Extended Horizon Adaptive Control
EPSAC	Extended Prediction Self Adaptive Control
E-NMPC	Economic Nonlinear Model Predictive Control
GMC	Generic Model Control
GPC	Generalized Predictive Control
IMC	Internal Model Control
ISM	Integral Sliding Mode
ISS	Input-to-State Stability
LMI	Linear Matrix Inequality
LP	Linear Programming
LPV	Linear Parameter Varying
LRMPC	Linearized Robust Model Predictive Controller
LRMPC-δu	incremental Linearized Robust Model Predictive Control
LRMPC-ISM	Linearized Robust Model Predictive Control- Integral Sliding Mode
LRMPC-PI	Linearized Robust Model Predictive Control-Proportional Integral
MAC	Model Algorithmic Control
MIMO	Multiple Input Multiple output
MPC	Model Predictive Control

NLP	NonLinear Programming
NMPC	Nonlinear Model Predictive Control
NMPC-$\varepsilon^{s/m}$	Nonlinear Model Predictive Control with $\varepsilon^{s/m}$ signal
NMPC-$j\varepsilon^{s/m}$	Nonlinear Model Predictive Control with $j\varepsilon^{s/m}$ signal
ODE	Ordinary Differential Equation
OLS	Ordinary Least Squares
PDE	Partial Differential Equation
PFC	Predictive Functional Control
PI	Proportional Integral
PVC	PolyVinyl Chloride
QIH-NMPC	Quasi-Infinite Horizon Nonlinear Model Predictive Control
QP	Quadratic Programming
RHC	Receding Horizon Control
RLS	Robust Least Squares
RNMPC	Robust Nonlinear Model Predictive Control
rRNMPC	reduced Robust Nonlinear Model Predictive Control
RTO	Real Time Optimization
SISO	Single Input Single output
SMC	Sliding Mode Control
SMPC	Stochastic Model Predictive Control
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicule
ZOH	Zero Order Hold

Preliminaries

Notations, basic definitions and properties are introduced and will be further used in the next chapters.

Notations

***Notation* 1.** Let $\mathbb{N}, \mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{Z}$ and $\mathbb{Z}_{\geq 0}$ denote natural, real, non-negative real, integer and non-negative integer number sets, respectively.

***Notation* 2.** $\mathbb{0}_{n\times m} \in \mathbb{R}^{n\times m}$ is the zero matrix of dimension $n \times m$ and $\mathbb{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix of dimension $n \times n$.

***Notation* 3.** The notation A^* denotes the conjugate transpose of the matrix A.

***Notation* 4.** The notation $A^\dagger$ denotes the pseudo inverse of the matrix A such that

$$A^\dagger \triangleq \lim_{\delta\to 0} \left(A^*A + \delta\mathbb{I}\right)^{-1} A^*$$

***Notation* 5.** Given $n \in \mathbb{Z}_{\geq 0}$, an arbitrary norm of a vector $x \in \mathbb{R}^n$ is denoted as $|x|$.

***Notation* 6.** $||z||_P^2 = z^\top Pz$ is the Euclidean norm weighted by the matrix P.

***Notation* 7.** Matrix norm $||A||$ is given by $||A|| = \sqrt{\bar{\sigma}(A^*A)}$ with $\bar{\sigma}(A)$ the maximum eigenvalue of A.

***Notation* 8.** The signum function of a real number x (denoted sign) is defined as follows:

$$\text{sign(x)} := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{1}$$

Basic definitions

***Definition* 1.** A symmetric $n \times n$ real matrix A is said to be positive semidefinite if the scalar $z^\top Az$ is non-negative for every non-zero column vector z of n real numbers. It is denoted $A \succ 0$

***Definition* 2.** A symmetric $n \times n$ real matrix A is said to be positive definite if the scalar $z^\top Az$ is positive for every non-zero column vector z of n real numbers. It is denoted $A \succeq 0$

***Definition* 3.** A matrix $A \in \mathbb{R}^{m \times n}$ is full column rank if and only if $A^\top A$ is invertible.

***Definition* 4.** Given an affine space A, a set $B \subseteq A$ is said to be convex if $\forall x, y \in B, \forall t \in [0,1] : (1-t)x + ty \in B$.

***Definition* 5.** Let A be a convex set and let $f : A \longrightarrow \mathbb{R}$. f is said to be convex if $\forall a, b \in A, \forall t \in [0,1] : f(ta + (1-t)b) \leq tf(a) + (1-t)f(b)$.

***Definition* 6.** An optimization problem of the form

$$\begin{aligned} &\min_x \ f(x) \\ &\text{s.t.} \ \ g_i(x) \leq 0, \quad i = 1, \ldots, m \end{aligned} \tag{2}$$

is called convex if the objective function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex on $\mathbb{R}^n$ and the constraints $g_1, \ldots, g_m : \mathbb{R}^n \longrightarrow \mathbb{R}$ are convex.

***Definition* 7.** A function f is said to be positive definite if $f(x) > 0$ for all $x > 0$.

***Definition* 8.** A function f: $\mathbb{R} \longrightarrow \mathbb{R}$ is said to be $\mathcal{C}^n$ function with $n \in \mathbb{Z}_{\geq 0}$, if the first n derivatives $f^{'}(.), f^{''}(.), \ldots, f^{(n)}(.)$ all exist and are continuous with respect to their argument.

***Definition* 9.** A function $f(x)$ is said to be locally lipschitz with respect to its argument x if there exists a positive scalar L_{fx} (so-called Lipschitz constant) such that $|f(x_1) - f(x_2)| \leq L_{fx}|x_1 - x_2|$ for all x_1 and x_2 in a given region of x.

***Definition* 10.** A function $\alpha(.)$: $\mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ is a $\mathcal{K}$-function (or of class $\mathcal{K}$) if it is continuous, positive definite, strictly increasing and $\alpha(0) = 0$.

***Definition* 11.** A function $\beta(.)$: $\mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ is a $\mathcal{K}_\infty$-function if it is a $\mathcal{K}$-function and also $\beta(s) \longrightarrow \infty$ as $s \longrightarrow \infty$.

***Definition* 12.** A function $\gamma(.,.)$: $\mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{KL}$ if, for each fixed $t \geq 0$, $\gamma(.,t)$ is of class $\mathcal{K}$, for each fixed $s \geq 0$, is decreasing and $\gamma(s,t) \longrightarrow 0$ as $t \longrightarrow \infty$.

***Definition* 13.** Let us consider an autonomous system

$$x_{k+1} = f(x_k, w_k), k \geq 0, x_0 = \bar{x} \tag{3}$$

where $x_k \in \mathbb{X}$ is the state of the system, $w_k \in \mathcal{W}$ is the disturbance vector (s.t. $\mathbb{X}$ and $\mathcal{W}$ are compact set that contain the origin).
A function $V(.)$: $\mathbb{R}^n \longrightarrow \mathbb{R}_{\geq 0}$ is called a Lyapunov function for system (3), if there exist sets $\mathbb{X}$ and $\mathcal{K}_\infty$-functions τ_1, τ_2, τ_3 s.t.

$$\begin{array}{l} \tau_1(|x_k|) \leq V(x_k) \leq \tau_2(|x_k|) \\ \Delta V(x_k, w_k) = V(f(x_k, w_k)) - V(x_k) \leq -\tau_3(|x_k|) \end{array} \tag{4}$$

with $x_k \in \mathbb{X}$ and $w_k \in \mathcal{W}$

***Definition* 14.** Let us consider the following discrete-time nonlinear system given by:

$$x_{k+1} = f(x_k, w_k), \quad k \geq 0, \quad x_0 = \bar{x} \tag{5}$$

with

$$|w_k| \leq \gamma(|x_k|) + \mu \tag{6}$$

where $\gamma(.)$ is a $\mathcal{K}$-function and μ is a modelled bound of uncertainties.
A function $V(.)$: $\mathbb{R}^n \longrightarrow \mathbb{R}_{\geq 0}$ is called a robust Lyapunov function if $\exists$ $\mathcal{K}_\infty$-functions $\alpha_1, \alpha_2, \alpha_3$ and σ, and a $\mathcal{K}$-function ζ s.t.

$$\begin{array}{l} \alpha_1(|x_k|) \leq V(x_k) \leq \alpha_2(|x_k|) + \sigma(\eta) \\ V(f(x_k, w_k)) - V(x_k) \leq -\alpha_3(|x_k|) + \zeta(\eta) \end{array} \tag{7}$$

with $x_k \in \mathbb{X}$ and $w_k \in \mathcal{W}$

***Definition* 15.** A set $\Phi \subset \mathbb{R}^n$ is a robust positively invariant set for the system (5), if $f(x_k, w_k) \in \Phi$, $\forall x_k \in \Phi$ and $\forall w_k \in \mathcal{W}$.

***Definition* 16.** Every pair (x_e, u_e) satisfying $F(x_e, u_e) = 0$ is called a steady-state of the ODE $\dot{x}(t) = F(x, u)$. This means that a process is at steady state (x_e, u_e) if and only if it remains at the same state when the input u_e is applied.

***Definition* 17.** Consider two vector fields $f(x)$ and $g(x)$ in $\mathbb{R}^n$. The Lie bracket operation generates a new vector field:

$$[f, g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g \tag{8}$$

Then, higher order Lie brackets can be defined as follows:

$$\begin{cases} \mathrm{ad}_f g \triangleq [f, g] \\ \mathrm{ad}_f^2 g \triangleq [f, [f, g]] \\ \vdots \\ \mathrm{ad}_f^k g \triangleq [f, \mathrm{ad}_f^{k-1} g] \end{cases} \tag{9}$$

Chapitre 1

Résumé

La commande des systèmes non linéaires soumis à des contraintes physiques sur l'état et l'entrée prend de plus en plus d'importance aux yeux de la communauté de l'automatique. Les méthodes classiques de commande non linéaire telles que le retour d'état linéarisant et les approches fondées sur le formalisme de Lyapunov offrent des solutions élégantes. Malheureusement, le développement de telles lois de commande devient de plus en plus difficile de par la complexité croissante du modèle mathématique nécessaire à leur mise en œuvre. C'est pour cette raison que la commande prédictive non linéaire (CPNL, nonlinear Model Predictive Control, NMPC en anglais) est une excellente alternative dont la méthodologie de synthèse reste assez simple tout en prenant en considération les contraintes et les incertitudes du modèle.

Par ailleurs, la commande des systèmes complexes, fortement non linéaires, incertains tels que les bioprocédés, s'avère être une tâche très délicate. En effet, dans ce cas, les paramètres du modèle sont généralement connus uniquement avec un intervalle de confiance associé (déterminé à partir d'une procédure d'identification par exemple). Par conséquent, toute stratégie de commande et en particulier la commande prédictive doit être robustifiée pour compenser le manque d'information et/ou inexactitudes paramètres. Il existe deux alternatives principales pour la robustification de la commande prédictive : l'approche stochastique (théorie probabiliste) et l'approche déterministe (formulation d'un problème d'optimisation de type min-max fondé sur la théorie des jeux), qui apparaissent comme des approches coûteuses en termes de temps de calcul. Ainsi, le défi qui doit être relevé dans ce travail consiste à synthétiser une loi de commande prédictive robuste vis-à-vis des incertitudes paramétriques avec un temps de calcul raisonnable, pour permettre une implémentation en temps réel. En partant de la formulation min-max de la commande prédictive, de nouvelles structures sont développées pour satisfaire le compromis entre la robustesse et la charge de calcul,

avec une application dédiée aux bioprocédés.

Cette thèse est structurée comme suit. Le Chapitre 2, non repris et résumé ci-après, propose une introduction portant sur le contexte, les motivations, les contributions et les publications issues des résultats obtenus pendant ces travaux. Le Chapitre 3 présente un état de l'art sur les techniques de commande prédictive existantes. Par la suite, la commande prédictive non linéaire permettant la poursuite d'une trajectoire de référence pour un système non linéaire avec une formulation continue/discrète est développée dans le Chapitre 4. De plus, une variante de la CPNL y est proposée pour permettre de tenir compte de l'écart entre la prédiction du modèle et la sortie du système. Dans le Chapitre 5, la commande prédictive robuste vis-à-vis d'incertitudes paramétriques est formulée en un problème d'optimisation du type min-max. Cependant, cette formulation induit une complexité qui peut être élevée ainsi qu'une charge de calcul importante, notamment dans le cas de systèmes multivariables, avec un nombre de paramètres incertains élevé. Pour y remédier, les deux approches proposées dans ces travaux consistent à simplifier le problème d'optimisation min-max, via l'analyse de sensibilité du modèle vis-à-vis de ses paramètres et la linéarisation des sorties prédites autour des valeurs nominales des paramètres du modèle et de la séquence de commande de référence, respectivement. La deuxième approche débouche sur la résolution d'un problème d'optimisation scalaire. Une analyse de la stabilité du système en boucle fermée est également proposée. La prise en compte de contraintes de type inégalité sur les variables d'optimisation transforme le problème initial en un problème convexe à deux niveaux : un niveau haut avec une optimisation scalaire et un niveau bas avec un problème de programmation quadratique. Le Chapitre 6 détaille deux améliorations de la loi de commande proposée. En premier lieu, la linéarisation des sorties prédites autour des valeurs nominales des paramètres du modèle et de la séquence de commandes optimales obtenue à l'itération précédente. Cette modification permet de rendre la solution moins sensible aux bruits de mesure. En second lieu, une structure de commande hiérarchisée combinant la commande prédictive robuste linéarisée et une commande auxiliaire (proportionnel intégrateur ou mode glissant intégral) est développée afin d'éliminer toute erreur statique en suivi de trajectoire de référence. Le Chapitre 7 est dédié à l'application des lois de commande développées à un cas d'étude de commande de bioréacteur de culture de microorganismes. Enfin, le Chapitre 8 reprend et résume le contenu du manuscrit ainsi que les perspectives proposées dans la continuité de ce travail. Chacun de ces chapitres se trouve résumé ci-dessous.

1.1 Chapitre 3 : Commande prédictive - état de l'art et principales stratégies

Le but de ce chapitre est de donner une introduction sur l'une des stratégies les plus puissantes permettant de résoudre les problèmes de commande non linéaires. Nous présentons ci-dessous la formulation du problème de commande prédictive ainsi qu'un état de l'art non exhaustif des principales stratégies prédictives rencontrées dans la littérature.

1.1.1 Contexte

La théorie de la commande optimale non linéaire prend son essor dans les années 1950-1960 résultant du principe du maximum de Pontryagin [116] et de la programmation dynamique développée par Bellman [12]. Le principe de l'horizon fuyant, concept clé de la commande prédictive, a été proposé par Propoi [117]. La philosophie de l'horizon fuyant est relativement proche du problème de la commande optimale en temps minimal et de la programmation linéaire [154]. Dans les années 70, la commande prédictive, grâce au progrès des méthodes de résolution, est devenue populaire dans l'ingénierie. Richalet *et al.* [129] et Cutler *et al.* [40] ont été les premiers à proposer l'application de la commande prédictive linéaire dans l'industrie. En général, beaucoup de systèmes sont fortement non linéaires. Dans ce cas, l'utilisation d'un modèle linéaire est souvent insuffisante pour bien décrire la dynamique du système de sorte qu'un modèle non linéaire doit être utilisé. Divers travaux sur la commande prédictive non linéaire ont été proposés dans la littérature [31, 77, 102, 113, 2, 95, 74]. La commande prédictive a impacté significativement le monde des techniques de régulation industrielle. Par conséquent, de nombreuse applications se retrouvent dans différents domaines : robotique [83], anesthésie clinique [89], industrie du ciment et usines de pâte à papier [132], tours de séchage et bras de robot [35], colonnes de distillation [70, 126], systèmes biochimiques [7], secteur pétrochimique [58], aérospatial [67, 21, 39], automobile [124], *etc.*

1.1.2 Principes de la commande prédictive

Soit le système discret non linéaire suivant :

$$x_{k+1} = f(x_k, u_k) \tag{1.1}$$

oú $x_k \in \mathbb{R}^{n_x}$ est l'état du système et $u_k \in \mathbb{R}^{n_u}$ est l'entrée de commande.
Étant donné le système (1.1), une formulation générale du problème de com-

mande prédictive non linéaire est donnée par

$$\min_{u_k,\ldots,u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} \phi(\hat{x}_{k+i}, u_{k+i-1}) \tag{1.2}$$

$$\text{s.c} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{pour} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k \end{cases} \tag{1.3}$$

où N_p est l'horizon de prédiction, $\hat{x}_{k+i}$ est l'état prédit, x_k est la condition initiale correspondant à l'état du système à l'instant k et $\phi(.,.)$ est le coût instantané (généralement un terme quadratique).

La commande prédictive non linéaire est implémentée selon le principe de l'horizon fuyant. Á l'instant k, l'état du système x_k est utilisé comme condition initiale et le problème d'optimisation (1.2)-(1.3) est résolu sur un horizon de prédiction de longueur N_p. Néanmoins, seule la première valeur de la séquence de commande est appliquée au système, *i.e.* u_k. La procédure est répétée à la période d'échantillonnage suivante. La structure de base globale d'une boucle de régulation prédictive est représentée sur la figure 1.1.

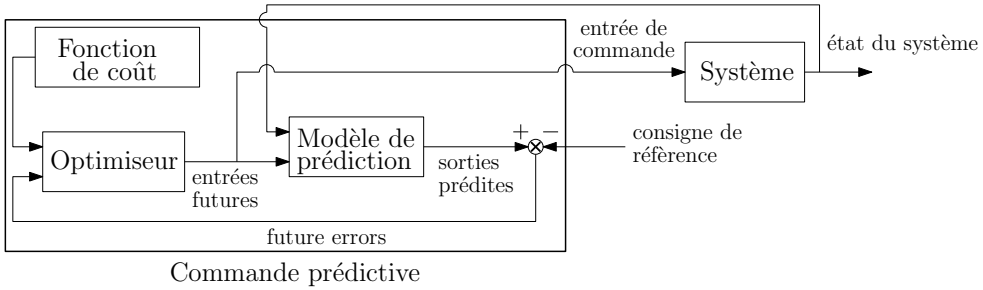

FIGURE 1.1 – Schéma de la structure générale de la commande prédictive.

Tous les algorithmes de commande prédictive ont les éléments communs suivants.

1.1.2.1 Modèle de prédiction

Un modèle du système est une représentation mathématique de la réalité. Il vise à prévoir l'évolution temporelle de chaque variable d'état et chaque sortie du système. En général, les modèles sont liés à la fois aux paramètres, les signaux d'entrée et de sortie. L'utilisation explicite du modèle (1.3) est due à la nécessité de calculer les états prédits $\hat{x}_{k+i}$ sur un horizon de prédiction déterminé de longueur N_p. Ces états prédits dépendent des valeurs connues jusqu'à l'instant k (*i.e.* x_k) et de la séquence de commandes futures, u_{k+i}, $i = \overline{0, N_p - 1}$, qui sont les variables d'optimisation.

1.1.2.2 Fonction de coût

Le critère qui doit être optimisé est une fonction quadratique qui peut prendre par exemple la forme suivante :

$$\sum_{i=1}^{N_p} \phi(\hat{x}_{k+i}, u_{k+i-1}) = \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + \sum_{i=1}^{N_c} ||\Delta u_{k+i-1}||^2_{\beta_i} \tag{1.4}$$

avec $\Delta u_{k+j} = 0, \quad j = \overline{N_c, N_p - 1}$ et $\Delta u_{k+i-1} = u_{k+i-1} - u_{k+i-2}$, et où $\hat{x}$ est la prédiction du modèle, x^r la référence sur la fenêtre de prédiction et Δu l'effort de commande nécessaire pour atteindre l'objectif spécifié. Le critère considère généralement les paramètres d'optimisation suivants :

- N_p et N_c sont les horizons de prédiction et de commande respectivement, qui ne sont pas nécessairement égaux ($N_c \leq N_p$).
- x^r est l'état de référence utilisé pour spécifier le comportement en boucle fermée et les performances de suivi attendues.
- $\alpha_i \in \mathbb{R}^{n_x \times n_x}$ et $\beta_i \in \mathbb{R}^{n_u \times n_u}$ sont les matrices de pondération sur l'erreur de suivi pour l'état et sur la commande respectivement (ici l'incrément de commande).

Les paramètres de réglage doivent être ajustés afin d'éviter l'instabilité du système et d'avoir de bonnes performances en boucle fermée.

De plus, des contraintes peuvent être prises en compte dans le problème d'optimisation. Par exemple, les actionneurs peuvent avoir un champ d'action limité. En ajoutant ces contraintes au problème d'optimisation, ce dernier devient plus complexe, de sorte que la solution ne peut pas être obtenue de manière explicite comme dans le cas sans contrainte. En outre, dans ce cas, le paradigme lié à l'obtention d'une solution faisable au problème d'optimisation doit être abordé.

1.1.2.3 Calcul de la loi de commande

L'optimiseur est un élément fondamental de la stratégie CPNL car il fournit les actions de commande en minimisant la fonction de coût (1.4). Une solution analytique peut être obtenue dans le cas d'un critère quadratique si le modèle est linéaire et qu'il n'y a pas de contraintes. Dans le cas contraire, la séquence de commande est déterminée à partir d'une stratégie d'optimisation en temps réel, impliquant la résolution d'un problème de programmation non linéaire. La taille du problème d'optimisation dépend du nombre de variables et de l'horizon de prédiction considéré. La démarche de résolution du problème d'optimisation peut être formalisée par les étapes suivantes :

1. Obtention des mesures
2. Calcul des prédictions via le modèle du système sur un certain horizon de prédiction
3. Calcul de la séquence de commandes optimales par minimisation de la fonction de coût
4. Implémentation de la première commande au système
5. Retour à l'étape 1.

1.1.3 État de l'art

En général, la commande prédictive peut être divisée en deux classes méthodologiques : linéaire et non linéaire. Maciejowski et Camacho donnent un bon aperçu de la théorie linéaire [90], tandis que Allgöwer et Zheng donnent un aperçu des méthodes non linéaires [3]. Les différents algorithmes de la commande prédictive résultent de la façon dont la fonction de coût à minimiser et les contraintes du système sont spécifiées.

1.1.3.1 Cas linéaire

En général, La Commande Prédictive Linéaire (CPL) utilise un modèle linéaire de faible complexité pour représenter le système. En présence de contraintes linéaires, le problème d'optimisation à résoudre est de taille raisonnable , ce qui peut être fait assez rapidement à chaque période d'échantillonnage afin d'être mis en œuvre dans le cadre de l'horizon fuyant. Dans ce cas, la solution du problème d'optimisation est la solution d'un problème de programmation linéaire ou quadratique (LP/QP), qui sont connus pour être convexes, et pour lesquels il existe une variété de méthodes et de logiciels numériques.

Selon les différentes stratégies de modélisation et de formulation du problème, de nombreuses variantes de la CPL existent dans la littérature : Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control (EHAC), Generalized Predictive Control (GPC).

1.1.3.2 Cas non linéaire

Du fait des limitations de la commande prédictive linéaire vis-à-vis des procédés ayant une dynamique fortement non linéaire, soumis à des contraintes

et/ou régis par un changement fréquent de régimes de fonctionnement, l'application de la commande prédictive non linéaire (CPNL) est à privilégier. L'utilisation d'un modèle non linéaire pour la prédiction transforme le problème quadratique convexe en un problème non convexe, plus difficile à résoudre. En conséquence, la convergence vers l'optimum global va fortement dépendre de l'étape d'initialisation. La CPNL est une méthode fondée sur l'optimisation pour la commande par retour d'état des systèmes non linéaires. Ses principales applications sont la stabilisation et les problèmes de suivi de trajectoires de référence. La CPNL est fondée comme la CPL sur le principe de l'horizon fuyant, où un problème de commande optimale en boucle ouverte sur un horizon fini est résolu à chaque instant d'échantillonnage et la séquence de commande optimale est appliquée jusqu'à ce qu'une nouvelle séquence de commande optimale soit disponible à l'instant suivant d'échantillonnage. Sa philosophie est donc similaire à celle de la CPL. Les méthodes considérées comme les plus représentatives sont les suivantes : CPNL à horizon infini, CPNL à horizon fini avec contrainte terminale d'égalité, CPNL à horizon fini avec contrainte terminale d'inégalité, CPNL à horizon fini avec coût terminal, CPNL à horizon quasi-infini, CPNL à retour d'état/sortie, CPNL économique.

1.1.4 Architectures de commande prédictive

- **Commande centralisée** : le régulateur gère tous les sous systèmes. La figure 1.2 est une représentation schématique de l'architecture centralisée de la commande prédictive (avec y^r la référence, y la sortie, u l'entrée de commande et x le vecteur d'état).

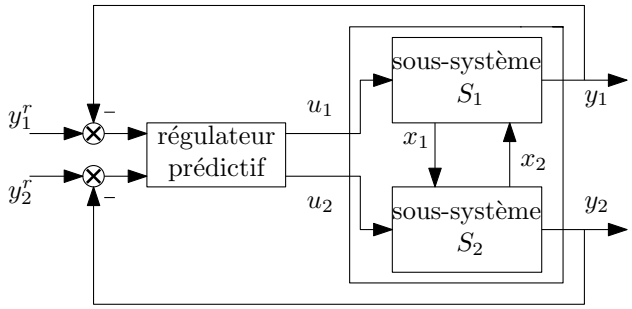

FIGURE 1.2 – Architecture de la commande prédictive centralisée [34].

- **Commande décentralisée** : ce type de stratégie est fondé sur une architecture de commande où l'entrée de commande et les variables

commandées sont regroupées dans des ensembles disjoints. Les régulateurs locaux sont synthétisés afin de fonctionner de façon totalement indépendante, comme l'exemple simple représenté sur la figure 1.3.

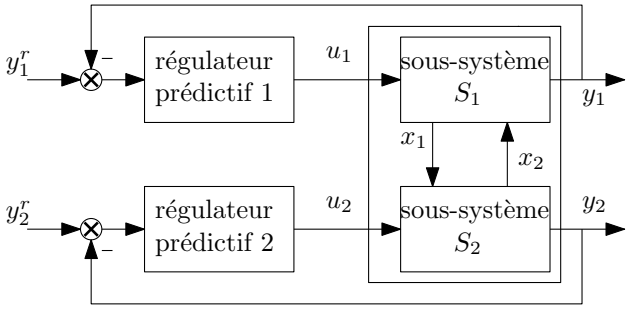

FIGURE 1.3 – Architecture de la commande prédictive décentralisée [34].

- **Commande distribuée** : La régulation globale est obtenue par la coopération de nombreux régulateurs, où chacun calcule un sous-ensemble d'entrées de commande individuellement avec d'éventuels échanges d'informations avec les autres régulateurs. En fait, les régulateurs locaux sont issus d'une synthèse prédictive, les informations transmises sont généralement les séquences de commandes prédites localement, de sorte que n'importe quel régulateur peut prédire les effets d'interaction sur l'horizon de prédiction considéré. Un exemple de stratégie distribuée est illustré figure 1.4.

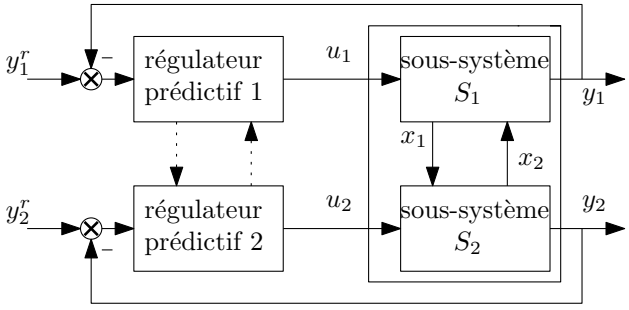

FIGURE 1.4 – Architecture de la commande prédictive distribuée [34].

Plusieurs approches ont été proposées dans la littérature. Elles diffèrent les unes des autres par les hypothèses faites sur le type d'interactions entre les différentes composantes de l'ensemble du système (algorithmes totalement ou partiellement connectés), le modèle d'échange d'informations entre les sous-systèmes, et la technique de conception du régulateur utilisé localement.

1.1.5 Robustesse

La CPNL est souvent utilisée dans des problèmes de commande. Cependant, les hypothèses de nominalité réduisent son utilisation ou ses performances. Par conséquent, d'autres types de commande prédictive sont développés pour tenir compte des incertitudes liées au modèle et des perturbations agissant sur ce système. Une large gamme de formulations qui incluent la robustesse dans la formulation du problème d'optimisation, existe dans la littérature. On distingue quatre types d'approches principales :

- Formulations de type LPV (Linear Parameter Varying) [6, 29, 148].
- Min-max NMPC [85, 93, 102, 88, 91, 93]
- H_∞-NMPC [60, 150, 151].
- Commande prédictive stochastique [28, 63].

1.1.6 Stabilité

L'une des questions clés de la commande prédictive est certainement la stabilité du système en boucle fermée. En effet, après avoir défini les principes théoriques de base de la commande prédictive, des sujets plus avancés tels que la stabilité robuste en présence de perturbations, l'estimation des performances et l'efficacité des algorithmes numériques ont été abordés dans la littérature. La stabilité de la commande prédictive pour les systèmes non linéaires contraints nécessite le recours à la théorie de la stabilité au sens de Lyapunov, qui peut être exprimée commodément par le biais des fonctions dites fonctions de comparaison, introduites dans la théorie de la commande non linéaire par Sontag [139]. La stabilité du système sans perturbation est appelée stabilité nominale, tandis que la stabilité du système en présence de perturbations est appelée stabilité robuste.

1.1.6.1 Stabilité nominale

La stabilité nominale d'un système en boucle fermée est prouvée par l'existence d'une fonction de Lyapunov pour le système. Les stratégies qui garantissent la stabilité nominale rencontrées dans la littérature sont : CPNL à horizon infini, CPNL à horizon fini avec contrainte terminale d'égalité, CPNL à horizon fini avec contrainte terminale d'inégalité, CPNL à horizon fini avec coût terminal, CPNL à horizon quasi-infini.

1.1.7 Stabilité robuste

Comme indiqué précédemment, la conception de lois de commande prédictive robuste se fait en prenant en compte les incertitudes d'une manière explicite afin d'optimiser la fonction objectif pour la pire configuration des incertitudes. Il existe de nombreuses approches permettant d'analyser la stabilité robuste, comme la structure entrée-état (ISS), la marge de stabilité robuste et la théorie des ensembles invariants couplée au cadre de la stabilité ISS-Lyapunov [85].

1.1.8 Avantages et inconvénients

La commande prédictive présente une série d'avantages en comparaison avec les autres lois de commandes :

- Formulation simple, basée sur des concepts bien compris et maitrisés.
- Méthodologie permettant des extensions futures.
- Applicabilité à une grande variété de systèmes, y compris les systèmes non linéaires et des systèmes à retard.
- Preuve de la stabilité pour les systèmes linéaires et non linéaires avec des contraintes d'entrée et d'état, sous certaines conditions spécifiques.
- Très utile quand les références futures sont connues a priori.
- Cas multivariable pouvant être facilement traité.
- Extension pour la prise en compte des contraintes (limitation des actionneurs).
- Utilisation explicite du modèle.
- Facilité de maintenance et de mise en œuvre en cas de changement du modèle ou de ses paramètres.

Cependant, la commande prédictive présente certains inconvénients :

- la commande prédictive nécessite généralement un temps de calcul plus important par rapport aux stratégies de commande classiques. La prise en compte des contraintes de type inégalité augmente encore plus la charge de calcul.
- En général, elle induit un manque de preuve de stabilité (sauf cas particuliers).

Cependant, la plus grande exigence est la nécessité d'un modèle approprié. En effet, la stratégie de commande est fondée sur une connaissance préalable du modèle, mais ne dépend pas d'une structure de modèle spécifique. Il est cependant évident que les avantages obtenus seront affectés par l'écart existant entre le processus réel et le modèle de prédiction.

1.2 Chapitre 4 : Commande prédictive non linéaire

Dans ce chapitre, on propose la formulation et la mise en œuvre d'une stratégie de commande prédictive non linéaire permettant la poursuite d'une trajectoire de référence.

1.2.1 Formulation du problème d'optimisation

Soit le système non linéaire à temps continu suivant :

$$\begin{cases} \dot{x}(t) = F(x(t), u(t), \theta), \quad x_0 = \bar{x} \\ y(t) = Hx(t) \end{cases} \tag{1.5}$$

où $x \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ est le vecteur d'état avec $\mathbb{X}$ l'ensemble compact des états admissibles, $\bar{x}$ est le vecteur d'état initial, $y \in \mathbb{Y} \subseteq \mathbb{R}^{n_y}$ est le vecteur des sorties mesurées avec $\mathbb{Y}$ l'ensemble compact des sorties admissibles et $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ est le vecteur des entrées de commande avec $\mathbb{U}$ l'ensemble compact des commandes admissibles. $\theta \in \mathbb{R}^{n_\theta}$ est le vecteur des paramètres incertains qui sont supposés se trouver dans l'ensemble compact $\Theta = [\theta^-, \theta^+]$ défini comme suit :

$$\theta = \theta_{\text{nom}} + \delta\theta \tag{1.6}$$

où θ_{nom} est le vecteur des paramètres nominaux défini comme étant la valeur moyenne (barycentre) de l'ensemble compact :

$$\theta_{\text{nom}} = \frac{\theta^+ + \theta^-}{2} \tag{1.7}$$

et $\delta\theta$ le vecteur des incertitudes paramétriques. L'application $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \longrightarrow \mathbb{R}^{n_x}$, de classe $\mathcal{C}^2$ vis-à-vis de tous ces arguments, représente la dynamique du système non linéaire. La matrice d'observation est donnée par $H \in \mathbb{R}^{n_y \times n_x}$.

Les sorties discrétisées sont obtenues à chaque période d'échantillonnage T_s constante par l'intégration du modèle d'état continu (1.5) en utilisant par exemple la méthode de *Runge-Kutta* avec un pas d'intégration T_d inférieur

à T_s. Ainsi le modèle de prédiction est défini par les équations récursives suivantes :

$$\begin{cases} x_{k+1} = f(x_k, u_k, \theta), k \geq 0, \quad x_0 = \bar{x} \\ y_k = Hx_k \end{cases} \tag{1.8}$$

où x_{k+1} est l'état à l'instant t_{k+1}, k est l'indice de temps, x_k et y_k sont le vecteur d'état et des mesures discrétisées à l'instant t_k, respectivement.
L'entrée de commande $u(t)$ est paramétrée en utilisant une approximation constante par morceaux sur un intervalle de temps $[t_k, t_{k+1}] \triangleq [kT_s, (k+1)T_s)]$:

$$u_k \triangleq u(\tau) = u(t_k), \quad \tau \in [t_k, t_{k+1}[\tag{1.9}$$

Soit la trajectoire d'état discrète g, définie comme étant la solution, à l'instant t_{k+1}, du système (1.5) :

$$\begin{cases} x_{k+1} = g(t_0, t_{k+1}, \bar{x}, u_{t_0}^{t_k}, \theta) \\ y_k = Hx_k \end{cases} \tag{1.10}$$

avec l'état initial x_0, et $u_{t_0}^{t_k}$ la séquence de commande à partir de l'instant initial t_0 jusqu'à l'instant t_k. La méthode utilisée pour obtenir g est supposée être la même que celle utilisée pour obtenir (1.8).
Il apparait clairement que :

$$f(x_k, u_k, \theta) \equiv g(t_k, t_{k+1}, x_k, u_k, \theta) \tag{1.11}$$

Par la suite, le modèle (1.10) sera utilisé dans la stratégie CPNL afin de prédire le comportement futur du système.

Dans cette étude, le problème CPNL est formulé à des fins de suivi de trajectoire. L'objectif principal est de contraindre la sortie y à suivre la trajectoire de référence y^r tandis que l'entrée de commande est contrainte à suivre la consigne u^r. En outre, les saturations sur les vecteurs d'états et d'entrée de commande avec des seuils minimal et maximal $x_{\min}$, $x_{\max}$, $u_{\min}$ et $u_{\max}$, respectivement peuvent être prises en considération (*i.e.* $\mathbb{X} = [x_{\min}, x_{\max}]$ et $\mathbb{U} = [u_{\min}, u_{\max}]$). Ces contraintes inégalité peuvent résulter de contraintes physiques et opérationnelles sur le système commandé.

Le contrôleur prédictif prédit l'évolution future du processus $\hat{y}_{k+1}^{k+N_p}$ sur un horizon temporel fini de longueur $N_p T_s$, en utilisant un modèle dynamique non linéaire. À chaque instant t_k, la séquence de commande optimale sur l'horizon de prédiction est obtenue en minimisant une fonction coût quadratique fondée sur les erreurs de suivi tout en assurant que les contraintes sont respectées. Seule la première valeur de la séquence optimale est appliquée au système, et toute la procédure est répétée à l'instant d'échantillonnage suivant selon le principe de l'horizon fuyant [90].

La fonction coût à minimiser est définie comme étant la somme de deux fonctions quadratiques, en fonction des erreurs de suivi sur l'horizon fuyant :

$$\Pi_{\text{NMPC}}(x_k, u_k^{k+N_p-1}) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \tag{1.12}$$

Le modèle de prédiction est donné par :

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta) \\ \hat{y}_{k+j} = H\hat{x}_{k+j}, \quad \forall j = \overline{1, N_p} \end{cases} \tag{1.13}$$

sous les contraintes suivantes reformulées matriciellement :

$$\begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \tag{1.14}$$

avec x_k le vecteur d'état à l'instant t_k,

$u_k^{k+N_p-1} = \begin{bmatrix} u_k \\ \vdots \\ u_{k+N_p-1} \end{bmatrix}$ le vecteur d'optimisation,

$u_k^{r,k+N_p-1} = \begin{bmatrix} u_k^r \\ \vdots \\ u_{k+N_p-1}^r \end{bmatrix}$ la séquence de commande de référence,

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, \theta_{\text{nom}}) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, \theta_{\text{nom}}) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, \theta_{\text{nom}}) \end{bmatrix} \text{ la séquence des sorties prédites,} \tag{1.15}$$

et $y_{k+1}^{r,k+N_p} = \begin{bmatrix} y_{k+1}^r \\ \vdots \\ y_{k+N_p}^r \end{bmatrix}$ les valeurs de consigne.

N_p est la longueur de l'horizon de prédiction. $V \succeq 0$ et $W \succ 0$ sont les matrices de pondération.

En supposant une parfaite connaissance du vecteur des paramètres θ (*i.e.* $\theta = \theta_{\text{nom}}$), la formulation du problème d'optimisation est traduite en un problème de programmation non linéaire sur un horizon de prédiction fini $N_p T_s$

à chaque instant d'échantillonnage. La séquence de commande optimale est obtenue en minimisant le critère de performance (1.12) avec prise en compte des contraintes (1.14) et de la dynamique du système comme suit :

$$\textbf{NMPC :} \qquad \overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \Pi_{\text{NMPC}}(x_k, u_k^{k+N_p-1}) \quad (1.16)$$

$$\text{s.c.} \quad (1.13)\text{-}(1.14)$$

1.2.2 Une variante de la CPNL

La prise en compte des erreurs induites par le modèle peut être assurée à partir d'une modification de la structure du problème d'optimisation par l'introduction du signal $\varepsilon^{s/m}$, qui permet de tenir compte de l'écart entre la prédiction du modèle et la sortie du système. La prise en compte de l'erreur, notée $\varepsilon_k^{s/m}$, entre la sortie du système $\hat{y}_{k+j|k}$ et celle du modèle $\hat{y}^m_{k+j|k}$, après j intervalles de prédiction, est donnée par la relation suivante :

$$\begin{cases} \hat{y}_{k+j|k} = \hat{y}^m_{k+j|k} + j\varepsilon_k^{s/m}, \quad j = \overline{1, N_p} \\ \varepsilon_k^{s/m} = y_k - \hat{y}^m_{k|k} \end{cases} \quad (1.17)$$

Le terme $j\varepsilon_k^{s/m}$ représente l'intégration de l'erreur de modélisation jusqu'à l'instant j. La structure de la CPNL avec prise en compte du signal erreur modèle/système est illustrée figure 1.5.

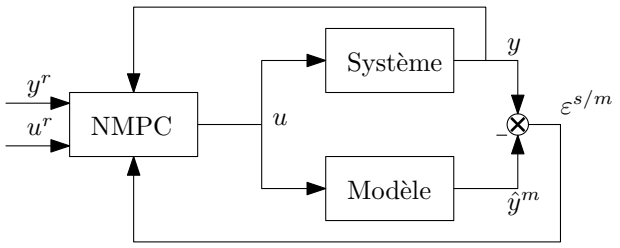

FIGURE 1.5 – CPNL incluant le signal $\varepsilon^{s/m}$.

On constate cependant que l'amélioration en termes de robustesse induite par cette structure reste limitée, nécessitant donc l'élaboration de véritables stratégies robustes.

1.3 Chapitre 5 : Commande prédictive non linéaire robuste

Ce chapitre s'attache à proposer une nouvelle stratégie de commande prédictive robuste. Elle est fondée sur l'approximation du problème min-max via la linéarisation du modèle de prédiction et l'utilisation de la dualité Lagrangienne. Dans ce contexte, deux méthodes ont été développées. Dans la première méthode, la séquence de commande optimale est calculée à partir d'un problème d'optimisation scalaire. La seconde méthode permet de prendre en considération explicitement les contraintes de type inégalité.

1.3.1 Stratégie min-max

On suppose que le vecteur des paramètres est incertain et appartient à l'ensemble Θ défini par (1.6)-(1.7). Ainsi, la robustification de la commande prédictive non linéaire induit la formulation d'un problème d'optimisation de type min-max [4, 93, 102]. La séquence de commande qui minimise la fonction coût est la solution du problème d'optimisation suivant :

$$\textbf{RNMPC :} \qquad \overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\theta} \Pi_{\text{RNMPC}}(x_k, u_k^{k+N_p-1}, \delta\theta) \quad (1.18)$$

soumis à

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta = \theta_{\text{nom}} + \delta\theta), \;\; j = \overline{1, N_p} \\ \theta \in \Theta \\ \begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \end{cases} \quad (1.19)$$

où θ et θ_{nom} sont donnés par (1.6) et (1.7). La fonction coût est donnée par :

$$\Pi_{\text{RNMPC}}(x_k, u_k^{k+N_p-1}, \delta\theta) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \quad (1.20)$$

Les sorties prédites $\hat{y}_{k+1}^{k+N_p}$ ont la mêmes formulation que dans (1.15) :

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, \theta) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, \theta) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, \theta) \end{bmatrix} \quad (1.21)$$

La séquence de commande optimale $\overset{\star}{u}_k^{k+N_p-1}$ est déterminée en minimisant l'erreur de suivi en considérant toutes les trajectoires possibles [54, 76].

Il apparait clairement que le temps de calcul augmente en fonction de la taille du vecteur des paramètres, du nombre d'entrée de commande et de la taille de l'horizon de prédiction pouvant vite devenir prohibitif. Ainsi, le défi principal est de réduire le temps de calcul tout en maintenant de bonnes performances en termes de précision de poursuite.

1.3.2 Commande prédictive robuste réduite

Étant donné que le problème d'optimisation min-max (1.18)-(1.20) est coûteux en termes de temps de calcul, il peut se simplifier en réduisant le nombre de paramètres θ incertains pris en compte dans l'optimisation à partir d'une analyse de sensibilité du modèle vis-à-vis de ses paramètres.

1.3.2.1 Analyse de sensibilité

La fonction de sensibilité $S_{\theta_j}^{x_i}$ représente la sensibilité de chaque état x_i aux faibles variations de chaque paramètre θ_j. Elle est donnée par l'expression suivante :

$$S_{\theta_j}^{x_i} \triangleq \frac{\partial x_i}{\partial \theta_j}, \quad i = \overline{1, n_x} \text{ et } j = \overline{1, n_\theta} \tag{1.22}$$

La dynamique des fonctions de sensibilité pour le système (1.5) est calculée comme suit :

$$\dot{S}_{\theta_j}^{x_i} = \frac{d}{dt}\left(\frac{\partial x_i}{\partial \theta_j}\right) = \frac{\partial}{\partial \theta_j}\left(\frac{dx_i}{dt}\right) = \frac{\partial F_i}{\partial \theta_j} + \sum_{k=1}^{n_x} \frac{\partial F_i}{\partial x_k}\left(\frac{\partial x_k}{\partial \theta_j}\right) \tag{1.23}$$

avec la condition initiale suivante : $\frac{\partial x_i}{\partial \theta_j} = 0$.
L'analyse de l'évolution temporelle des fonctions de sensibilité selon l'ordre de grandeur de leurs amplitudes, permet de sélectionner les paramètres κ qui ont le plus d'influence.

1.3.2.2 Reformulation du problème

Par la suite, seuls les paramètres les plus influents vont être considérés dans le problème d'optimisation min-max, au lieu de tous les paramètres (avec $\theta \triangleq [\kappa, \zeta]$). Les paramètres influents κ sont définis par :

$$\begin{cases} \kappa = \kappa_{\text{nom}} + \delta\kappa & (1.24) \\ \kappa_{\text{nom}} = \dfrac{\kappa^+ + \kappa^-}{2} & (1.25) \end{cases}$$

où κ_{nom} sont les valeurs nominales des paramètres les plus influents et $\delta\kappa$ est le vecteur de leurs incertitudes. Les autres paramètres, ζ, sont maintenus à leurs valeurs nominales avec $\zeta_{nom} = (\zeta^+ + \zeta^-)/2$.

Grâce à l'analyse de sensibilité, le problème d'optimisation min-max (5.1)-(5.3) est remplacé par le problème suivant :

$$\textbf{rRNMPC :} \quad \overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\kappa} \Pi_{\text{rRNMPC}}(x_k, u_k^{k+N_p-1}, \delta\kappa) \quad (1.26)$$

soumis à

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \begin{bmatrix} \kappa_{\text{nom}} + \delta\kappa \\ \zeta_{nom} \end{bmatrix}), \;\; j = \overline{1, N_p} \\ \kappa \in \Theta_\kappa \\ \begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \end{cases} \quad (1.27)$$

La nouvelle fonction coût est définie par :

$$\Pi_{\text{rRNMPC}}(u_k^{k+N_p-1}, \delta\kappa) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \quad (1.28)$$

Les sorties prédites $\hat{y}_{k+1}^{k+N_p}$ sont données par :

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \end{bmatrix} \quad (1.29)$$

Néanmoins, cette approche ne peut pas être appliquée pour les systèmes dont les paramètres ont une influence similaire, aucune réduction des paramètres n'est alors possible. Par conséquent, cette approche ne peut pas être utilisée dans tous les cas.

1.3.3 Commande prédictive robuste linéarisée

Dans le but de réduire le temps de calcul induit par la résolution du problème d'optimisation de type min-max, l'approche proposée ici consiste à linéariser les sorties prédites. En conséquence, le problème non convexe initial est approché par un problème convexe plus facile à résoudre.

1.3.3.1 Principe général

La trajectoire d'état (1.11) à l'instant t_{k+j} est linéarisée (développement en série de Taylor limité au premier ordre) autour des paramètres nominaux θ_{nom} et de la séquence de commande de référence $u_k^{r,k+j-1}$, pour $j = \overline{1, N_p}$:

$$\hat{x}_{k+j} \approx \hat{x}_{\text{nom},k+j} + \nabla_\theta g(t_{k+j})\delta\theta + \nabla_u g(t_{k+j}) \left(u_k^{k+j-1} - u_k^{r,k+j-1} \right) \tag{1.30}$$

avec

$$\begin{cases} \hat{x}_{\text{nom},k+j} = g(t_k, t_{k+j}, x_k, u_k^{r,k+j-1}, \theta_{\text{nom}}) & (1.31) \\ \nabla_\theta g(t_{k+j}) = \left. \dfrac{\partial g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta)}{\partial \theta} \right|_{\substack{u_k^{k+j-1} = u_k^{r,k+j-1} \\ \theta = \theta_{\text{nom}}}} & (1.32) \\ \nabla_u g(t_{k+j}) = \left. \dfrac{\partial g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta)}{\partial u_k^{k+j-1}} \right|_{\substack{u_k^{k+j-1} = u_k^{r,k+j-1} \\ \theta = \theta_{\text{nom}}}} & (1.33) \end{cases}$$

Ainsi, les sorties prédites sont données par l'expression suivante :

$$\hat{y}_{k+1}^{k+N_p} = G_{\text{nom},k+1}^{k+N_p} + G_{\theta,k+1}^{k+N_p}\delta\theta + G_{u,k}^{k+N_p-1}(u_k^{k+N_p-1} - u_k^{r,k+N_p-1}) \tag{1.34}$$

avec

$$G_{\text{nom},k+1}^{k+N_p} = \begin{bmatrix} H\hat{x}_{\text{nom},k+1} \\ \vdots \\ H\hat{x}_{\text{nom},k+j} \\ \vdots \\ H\hat{x}_{\text{nom},k+N_p} \end{bmatrix}, \; G_{\theta,k+1}^{k+N_p} = \begin{bmatrix} H\nabla_\theta g(t_{k+1}) \\ \vdots \\ H\nabla_\theta g(t_{k+j}) \\ \vdots \\ H\nabla_\theta g(t_{k+N_p}) \end{bmatrix}, \; G_{u,k}^{k+N_p-1} = \begin{bmatrix} H\nabla_u g(t_{k+1}) \\ \vdots \\ H\nabla_u g(t_{k+j}) \\ \vdots \\ H\nabla_u g(t_{k+Np}) \end{bmatrix}$$

En utilisant l'équation (1.34), la fonction coût initiale Π_{RNMPC} (1.20) est approchée par :

$$\begin{aligned} &\Pi(x_k, u_k^{k+N_p-1}, \delta\theta) \approx ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + \\ &||G_{\text{nom},k+1}^{k+N_p} - y_{k+1}^{r,k+N_p} + G_{\theta,k+1}^{k+N_p}\delta\theta + G_{u,k}^{k+N_p-1}(u_k^{k+N_p-1} - u_k^{r,k+N_p-1})||_W^2 \end{aligned} \tag{1.35}$$

Finalement, le problème d'optimisation (1.18) devient :

$$\overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\theta} \; \Pi(x_k, u_k^{k+N_p-1}, \delta\theta) \tag{1.36}$$

soumis à

$$\theta \in \Theta, \quad x \in \mathbb{X}, \quad u \in \mathbb{U} \tag{1.37}$$

1.3.3.2 Analyse de la stabilité

***Théorème* 1.1.** Soit le système non linéaire à temps discret suivant :

$$x_{k+1} = l(x_k, w_k), \quad k \geq 0, \quad x_0 = \bar{x} \tag{1.38}$$

où $x_k \in \mathcal{X}$ est l'état du système, $w_k \in \mathcal{W}$ est la perturbation (telle que $\mathcal{W}$ est un ensemble compact contenant l'origine).
Si le système (1.38) admet une fonction de Lyapunov robuste alors le système est robuste stable.

Preuve. voir [86].

La stabilité du système (1.8) avec (1.36)-(1.37) en boucle fermée est analysée en utilisant le théorème précédent.

Soit le système à l'instant t_{k+1} :

$$\begin{aligned} x_{k+1} = f(x_k, u_k, \theta_{\text{nom}} + \delta\theta^s) \triangleq & f(x_k, u_k^r, \theta_{\text{nom}}) + \nabla_u f(u_k^r, \theta_{\text{nom}}).(u_k - u_k^r) + \\ & \nabla_\theta f(u_k^r, \theta_{\text{nom}}).\delta\theta^s + \vartheta(|u_k - u_k^r|^2) + \vartheta(|\delta\theta^s|^2) \end{aligned} \tag{1.39}$$

où $\vartheta(|.|^2)$ est le reste du développement en série de Taylor.
Soit le modèle de prédiction à l'instant t_{k+1} :

$$\begin{aligned} \hat{x}_{k+1|k} = f_p(x_k, u_k, \theta_{\text{nom}} + \delta\theta) \triangleq & f(x_k, u_k^r, \theta_{\text{nom}}) + \nabla_u f(u_k^r, \theta_{\text{nom}}).(u_k - u_k^r) \\ & + \nabla_\theta f(u_k^r, \theta_{\text{nom}}).\delta\theta \end{aligned} \tag{1.40}$$

Après développement, on obtient une borne sur l'erreur de prédiction :

$$\exists \alpha \in \mathbb{R}^+, \forall x, \forall u, \forall \theta, \quad |\nabla_\theta f| \leq \alpha \tag{1.41}$$

et

$$\exists \eta \in \mathbb{R}^+, \text{ tel que } \eta = \max(|\vartheta(|u_k - u_k^r|^2) + \vartheta(|\delta\theta^s|^2)|, \max(|\delta\theta|, |\delta\theta^s|)) \tag{1.42}$$

alors

$$|x_{k+1} - \hat{x}_{k+1|k}| \leq (2\alpha + 1)\eta \triangleq \Lambda(\eta) \tag{1.43}$$

où Λ est une fonction $\mathcal{K}_\infty$.
La fonction coût optimale Π peut être bornée comme suit :

$$\alpha_\psi(|x_k|) \leq \Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \leq \beta_t(|x_k|) + \varphi(\eta) + N_p\chi(\eta) \tag{1.44}$$

où α_ψ, β_t, et χ sont des fonctions $\mathcal{K}_\infty$ et φ est une fonction $\mathcal{K}$.
De plus, la variation de la fonction coût $\Delta\Pi^\star$ est donnée par l'expression suivante :

$$\Delta\Pi^\star \triangleq \quad \Pi(x_{k+1}, \breve{u}, \overset{\star}{\delta\theta}) - \Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \leq -\alpha_\psi(|x|) + \bar{\chi}(\eta) \tag{1.45}$$

avec

$$\bar{\chi}(\eta) = \chi(\eta) + \left(L_{tx} L_{fx}^{N_p-1} + L_{\psi x} \sum_{j=0}^{N_p-2} L_{fx}^{j} \right) \Lambda(\eta) \tag{1.46}$$

où $\bar{\chi}$ est une fonction $\mathcal{K}_\infty$.

On montre que la fonction coût optimale Π est une fonction de Lyapunov robuste. Par conséquent, le système (1.8) commandé par $\pi(x) = \overset{\star}{u}_k$ est robuste stable.

1.3.4 Implémentation de la commande prédictive robuste linéarisée en absence de contraintes (LRMPC)

Le problème d'optimisation (1.36)-(1.37) est résolu en utilisant la dualité $\mathcal{L}$agrangienne [23]. On se ramène alors à un problème d'optimisation scalaire [135]. Il en résulte les étapes d'implémentation suivantes :

Étape 1. Le multiplicateur de Lagrange $\lambda^\star$ est calculé comme suit :

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p^\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(\lambda) \tag{1.47}$$

Le critère $\mathbb{J}(\lambda)$ est défini par l'expression suivante :

$$\mathbb{J}(\lambda) = ||z(\lambda)||_V^2 + \lambda||\delta\theta_{\max}||^2 + ||G_{u,k}^{k+N_p-1} z(\lambda) - y_{k+1}^{r,k+N_p} + G_{\text{nom},k+1}^{k+N_p}||_{W(\lambda)}^2 \tag{1.48}$$

avec

$$z(\lambda) = (V + G_{u,k}^{k+N_p-1^\top} W(\lambda) G_{u,k}^{k+N_p-1})^\dagger (G_{u,k}^{k+N_p-1^\top} W(\lambda)(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p})) \tag{1.49}$$

et

$$W(\lambda) = W + W G_{\theta,k+1}^{k+N_p} (\lambda \mathbb{I} - G_{\theta,k+1}^{k+N_p^\top} W G_{\theta,k+1}^{k+N_p})^\dagger G_{\theta,k+1}^{k+N_p^\top} W \tag{1.50}$$

Étape 2. La séquence de commande $\overset{\star}{u}_k^{k+N_p-1}$ est donnée par :

$$\begin{aligned} \overset{\star}{u}_k^{k+N_p-1} =& u_k^{r,k+N_p-1} + (V + G_{u,k}^{k+N_p-1^\top} W(\lambda^\star) G_{u,k}^{k+N_p-1})^\dagger \\ & \times (G_{u,k}^{k+N_p-1^\top} W(\lambda^\star)(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p})) \end{aligned} \tag{1.51}$$

avec $W(\lambda^\star)$ donnée par (1.50) pour $\lambda = \lambda^\star$.

1.3.5 Commande prédictive robuste linéarisée avec prise en compte des contraintes

La prise en compte des contraintes inégalité sur la commande dans le problème d'optimisation (1.36)-(1.37) conduit à la résolution d'un problème à deux niveaux, selon les étapes suivantes :

Étape 1. Le scalaire $\lambda^\star$ est calculé à partir du problème d'optimisation suivant :

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(z(\lambda), \lambda) \tag{1.52}$$

La fonction $\mathbb{J}(z(\lambda), \lambda)$ est donnée par (1.48) et $z(\lambda)$ est calculé par :

$$z(\lambda) = \arg \min_{\underline{z} \leq z \leq \overline{z}} \left[z^\top E(\lambda) z - 2B(\lambda)^\top z \right] \tag{1.53}$$

où

$$\begin{cases} E(\lambda) = V + {G_{u,k}^{k+N_p-1}}^\top W(\lambda) G_{u,k}^{k+N_p-1} & (1.54) \\ B(\lambda) = {G_{u,k}^{k+N_p-1}}^\top W(\lambda) \left(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} \right) & (1.55) \\ \underline{z} = u_{min} \mathbb{I}_{n_u} - u_k^{r,k+N_p-1} & (1.56) \\ \overline{z} = u_{max} \mathbb{I}_{n_u} - u_k^{r,k+N_p-1} & (1.57) \end{cases}$$

avec $W(\lambda)$ donnée par (1.50).

Étape 2. La séquence de commande $\overset{\star}{u}_k^{k+N_p-1}$ est calculée à partir du problème d'optimisation (1.53) pour $\lambda = \lambda^\star$:

$$\overset{\star}{u}_k^{k+N_p-1} = u_k^{r,k+N_p-1} + z(\lambda^\star) \tag{1.58}$$

1.4 Chapitre 6 : Améliorations de la commande prédictive robuste linéarisée

Ce chapitre propose deux améliorations de la loi de commande prédictive robuste linéarisée. En premier lieu, la linéarisation des sorties prédites est effectuée ici autour des valeurs nominales des paramètres du modèle et de la séquence de commandes optimales obtenues à l'instant d'échantillonnage précédent. Cette modification permet de rendre la solution moins sensible aux bruits de mesure. En second lieu, une structure de commande hiérarchisée combinant la commande prédictive robuste linéarisée et une commande auxiliaire (proportionnel intégrateur ou mode glissant intégral) est développée afin d'éliminer toute erreur statique en suivi de trajectoire de référence.

1.4.1 Commande prédictive robuste linéarisée avec incréments de commande

Afin de rendre la commande prédictive robuste linéarisée moins sensible aux bruits de mesure, par rapport au chapitre précédent, la fonction coût quadratique mesure les efforts de commande au lieu de la différence entre les entrées de commande et leurs valeurs de référence. L'idée est d'utiliser une linéarisation du modèle autour des paramètres nominaux et de la séquence de commande optimale obtenue au pas d'échantillonnage précédent. On constate que, cette approche présente l'avantage d'avoir une solution moins sensible aux bruits de mesure. Ainsi, le nouveau problème d'optimisation est le suivant :

$$\delta \overset{\star}{u}_k^{k+N_p-1} = \arg \min_{\delta u_k^{k+N_p-1} \in \mathcal{U}^\delta} \max_{\delta\theta \in \Theta^\delta} \Pi_\delta(x_k, \delta u_k^{k+N_p-1}, \delta\theta) \tag{1.59}$$

soumis à

$$\hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta = \theta_{\text{nom}} + \delta\theta), \;\; j = \overline{1, N_p} \tag{1.60}$$

où

$$\Pi_\delta(x_k, \delta u_k^{k+N_p-1}, \delta\theta) \triangleq ||\delta u_k^{k+N_p-1}||_V^2 + ||\hat{y}_k^{k+N_p} - y_k^{r,k+N_p}||_W^2 \tag{1.61}$$

Les variables d'optimisation $\delta u_k^{k+N_p-1}$ sont définies comme suit :

$$\delta u_k^{k+N_p-1} = \begin{bmatrix} u_k - \overset{\star}{u}_{k-1} \\ \vdots \\ u_{k+j} - u_{k+j-1} \\ \vdots \\ u_{k+N_p-1} - u_{k+N_p-2} \end{bmatrix} \triangleq \begin{bmatrix} \delta u_k \\ \vdots \\ \delta u_{k+j} \\ \vdots \\ \delta u_{k+N_p-1} \end{bmatrix} \tag{1.62}$$

avec $\overset{\star}{u}_{k-1}$ l'entrée de commande appliquée à l'instant $k-1$ (solution du problème d'optimisation (1.59) à l'instant $k-1$).
Les sorties prédites sont données par l'expression suivante :

$$\hat{y}_{k+1}^{k+N_p} = G_{\text{nom},k+1}^{k+N_p} + G_{\theta,k+1}^{k+N_p}\delta\theta + G_{u,k}^{k+N_p-1}(\Xi_k^{k+N_p-1} + T_{N_p}\delta u_k^{k+N_p-1}) \tag{1.63}$$

avec

$$\Xi_k^{k+N_p-1} = \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k-1} \end{bmatrix} - \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k+N_p-2} \end{bmatrix} \tag{1.64}$$

et $T_{N_p} \in \mathbb{R}^{N_p \times N_p}$ une matrice triangulaire inférieure unitaire.

En suivant la même démarche qu'à la section 1.3.3, on obtient la solution suivante (cas sans contraintes) :

Étape 1. Le scalaire $\lambda^\star$ est calculé à partir du problème d'optimisation suivant :

$$\lambda^\star = \arg \min_{\lambda \geq \|G_{\theta,k+1}^{k+N_p\top} W G_{\theta,k+1}^{k+N_p}\|} \mathbb{J}(z(\lambda), \lambda) \tag{1.65}$$

où $\mathbb{J}(z(\lambda), \lambda)$ et $W(\lambda)$ sont données par (1.48) et (1.50).
La solution $z(\lambda)$ est calculé comme-suit :

$$\begin{aligned} z(\lambda) = &\left(V + T_{N_p}^\top G_{u,k}^{k+N_p-1\top} W(\lambda) G_{u,k}^{k+N_p-1} T_{N_p}\right)^\dagger T_{N_p}^\top G_{u,k}^{k+N_p-1\top} W(\lambda) \\ &\times \left(\bar{y}_{k+1}^{k+N_p} - G_{\text{nom},k+1}^{k+N_p} - G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1}\right) \end{aligned} \tag{1.66}$$

Étape 2. La séquence de commande $\overset{\star}{u}{}_k^{k+N_p-1}$ est donnée par :

$$\overset{\star}{u}{}_k^{k+N_p-1} = \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k-1} \end{bmatrix} + T_{N_p} z(\lambda^\star) \tag{1.67}$$

1.4.2 Stratégie de commande hiérarchisée

Les approches prédictives robustes présentées précédemment peuvent avoir des problèmes de précision surtout pour une grande période d'échantillonnage du fait que l'on utilise une approximation du premier ordre des fonctions non linéaires. Dans cette partie, une stratégie de commande hiérarchisée telle que représentée sur la figure 1.6 est proposée.

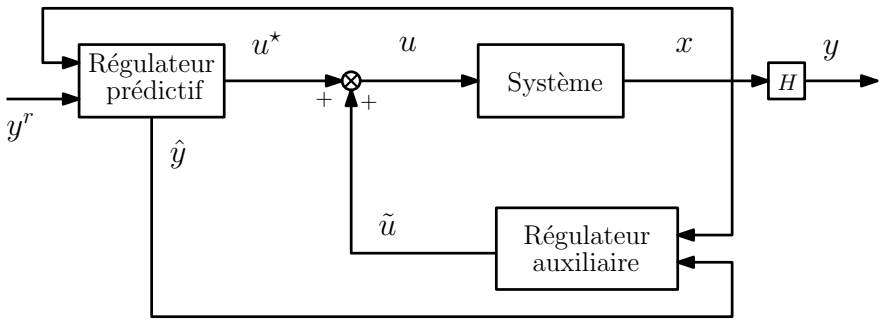

FIGURE 1.6 – Schéma de la stratégie de commande hiérarchisée.

La boucle de régulation est formée par une loi de commande prédictive robuste (entrée de commande notée $u^\star$) combinée à un correcteur auxiliaire (entrée de commande notée $\tilde{u}$). Le régulateur prédictif permet de suivre la trajectoire de référence, alors que le régulateur supplémentaire annule toute erreur de suivi résiduelle.

Deux types de régulation additionnelle sont étudiés : une structure proportionnelle-intégrale (PI) et une stratégie de commande par mode glissant intégrateur (ISM).

En premier lieu, la loi de commande proportionnel intégrateur est choisie pour sa simplicité. Elle est donnée par l'expression suivante :

$$\tilde{u}(t) = K_p(y(t) - \hat{y}(t)) + \frac{K_p}{T_i}\int_{t_{k-1}}^{t}(y(\tau) - \hat{y}(\tau))d\tau \tag{1.68}$$

où K_p et T_i sont les deux paramètres de réglage du PI et $\hat{y}$ la prédiction du modèle.

En second lieu, on considère une loi de commande comme régulateur auxiliaire. La loi de commande par mode glissant intégrateur présente l'avantage d'être robuste vis-à-vis des perturbations. La surface de glissement est choisie comme suit :

$$\phi(x,t) = Z_2(t) + \xi_3 Z_1(t) \tag{1.69}$$

avec

$$\begin{cases} Z_1(t) = \int_{t_{k-1}}^{t}(y(\tau) - \hat{y}(\tau))\,\mathrm{d}\tau \\ \\ Z_2(t) = \frac{1}{\xi_1}(y(t) - \hat{y}(t) - \xi_2 Z_1(t)) \end{cases} \tag{1.70}$$

L'attractivité de la surface de glissement est donnée ci-dessous :

$$\dot{\phi}(x,t) = -K_s \mathrm{sign}(\phi(x,t)),\ \ K_s \in \mathbb{R}^+ \tag{1.71}$$

Ainsi, l'expression de la loi de commande par mode glissant intégrateur (ISM) est la suivante :

$$\tilde{u}(t) = \frac{1}{\eta}(-\varphi - \chi u_{k-1} - \xi_1 K_s \mathrm{sign}(\phi(x,t)) + (\xi_2 - \xi_1\xi_3)(\xi_1 Z_2(t) + \xi_2 Z_1(t))) \tag{1.72}$$

avec

$$\begin{cases} \varphi = H(f_x(x(t),\theta) - f_x(\hat{x}(t),\theta_{nom})), \\ \chi = H(f_u(x(t),\theta) - f_u(\hat{x}(t),\theta_{nom})), \\ \eta = H f_u(x(t),\theta) \end{cases} \tag{1.73}$$

et

$$\begin{cases} \dot{x}(t) = f_x(x(t),\theta) + f_u(x(t),\theta)u(t), \quad \forall t > t_0, \quad x(t_0) = x_0 \\ y(t) = Hx(t) \end{cases} \tag{1.74}$$

Dans (1.73), θ est remplacé par θ_{nom} (ou par une estimée de θ) car θ est incertain.

1.5 Chapitre 7 : Application à un système de culture de microalgues

Le développement des technologies environnementales a pris une place de plus en plus importante dans l'industrie. En effet, grâce à leurs caractéristiques biochimiques, les bioprocédés sont considérés comme une alternative prometteuse aux sources d'énergie fossiles. Dans l'industrie des biotechnologies, la commande des bioréacteurs est devenue un enjeu majeur du fait que ce sont des systèmes fortement non linéaires et incertains par nature nécessitant la synthèse d'une stratégie de commande robuste. Dans ce chapitre, l'objectif principal est d'appliquer les lois de commande présentées précédemment à un système de culture de microalgue pour la régulation de la concentration en biomasse à une valeur de consigne en présence d'incertitudes paramétriques et de bruit de mesure.

1.5.1 Modélisation du système

Le modèle dynamique considéré suppose que le photobioréacteur est en mode continu (le débit moyen de soutirage est égal à celui d'alimentation, conduisant à un volume constant), sans aucune biomasse supplémentaire dans l'alimentation et en négligeant l'effet des échanges gazeux. Les bilans de matières conduisent aux équations différentielles suivantes :

$$\begin{cases} \dot{X}(t) = \mu(Q(t), I(t))\, X(t) - DX(t) \\ \dot{Q}(t) = \rho(S(t)) - \mu(Q(t), I(t))\, Q(t) \\ \dot{S}(t) = (S_{in} - S(t))D - \rho(S(t))X(t) \end{cases} \tag{1.75}$$

avec

$$\begin{cases} \mu(Q, I) = \bar{\mu}(1 - \dfrac{K_Q}{Q})\mu_I \quad \textbf{(Droop)} \\ \rho(S) = \rho_m \dfrac{S}{S + K_s} \quad \textbf{(Michaelis-Menten)} \\ \mu_I = \dfrac{I}{I + K_{sI} + \dfrac{I^2}{K_{iI}}} \quad \textbf{(Haldane)} \end{cases} \tag{1.76}$$

où X est la concentration en biomasse (μm^3 L^{-1}), Q le quota interne (μmol-μm^{-3}) et S la concentration en substrat (μmol L^{-1}). D (d^{-1}) est le taux de

dilution défini comme le ratio entre le débit et le volume (d^{-1}), $\mu(.)$ est la vitesse de croissance (d^{-1}), $\bar{\mu}$ le taux de croissance maximal, K_Q la quantité de quota interne en-dessous de laquelle les bactéries ne croissent plus (μmol μm^{-3}), $\rho(.)$ la vitesse d'absorption (μmol μm^{-3} d^{-1}), ρ_m le taux d'absorption maximal (μmol μm^{-3} d^{-1}) et K_s la constante de saturation par le substrat (μmol L^{-1}). La fonction μ_I représente l'effet de la lumière sur le taux de croissance ($-$), I est l'intensité lumineuse (μE m^{-2} s^{-1}), K_{sI} la constante de saturation par la lumière (μE m^{-2} s^{-1}) et K_{iI} la constante d'inhibition par la lumière (μE m^{-2} s^{-1}). Les paramètres du modèle (1.75) sont donnés dans le tableau suivant :

Tableau 1.1 – Paramètres du modèle.

Paramètre	Valeur	Unité
$\bar{\mu}$	2	d^{-1}
ρ_m	9,3	μmol μm^{-3} d^{-1}
K_Q	1,8	μmol μm^{-3}
K_s	0,105	μmol L^{-1}
K_{sI}	150	μE m^{-2} s^{-1}
K_{iI}	2000	μE m^{-2} s^{-1}

1.5.2 Stratégie de commande

L'objectif de commande est de réguler la concentration en biomasse à une valeur de consigne en agissant sur le taux de dilution tout en respectant les contraintes biologiques.

1.5.3 Résultats en simulation

Dans cette partie, les performances des stratégies de commande prédictives sont évaluées dans le cas de paramètres incertains et en présence de bruit de mesure. Les conditions de simulation sont résumées dans le tableau 1.2.
Les quatre lois de commande prédictives sont tout d'abord comparées : la commande prédictive non linéaire (NMPC), la formulation min-max robuste (RNMPC), la formulation min-max réduite (rRNMPC) et la commande prédictive robuste linéarisée (LRMPC) sans prise en compte des contraintes explicitement et saturation de la commande a posteriori. Grâce à l'analyse de sensibilité du modèle vis-à-vis de ses paramètres, il apparait que la constante de saturation par le substrat K_s et la quantité de quota interne minimale K_Q sont les paramètres qui ont le plus d'influence sur l'évolution de la concentration en biomasse ($\kappa = [K_s, K_Q]^\top$).

Tableau 1.2 – Conditions de simulation.

	Variable	Valeur	Unité
période d'échantillonnage	T_s	10	min
pas d'intégration	T_d	0.2	min
temps de simulation	T_f	1	d
substrat dans l'alimentation	S_{in}	100	μmol L^{-1}
intensité lumineuse optimale	I_{opt}	547	μE m^{-2} s^{-1}
quota maximum de cellules	Q_l	9	μmol μm^{-3}
taux de dilution maximal	$D_{\max}$	1,6	d^{-1}
horizon de prédiction	N_p	5	-
matrice de pondération sur l'entrée	V	$\mathbb{I}_{N_p}$	-
matrice de pondération sur l'état	W	$\mathbb{I}_{N_p}$	-
concentration initiale en biomasse	$X(0)$	24,95	μm^3 L^{-1}
quota interne initial	$Q(0)$	4	μmol μm^{-3}
concentration initiale en substrat	$S(0)$	0,05	μmol L^{-1}

Les mesures de la concentration en biomasse sont entachées d'un bruit blanc gaussien de moyenne nulle et d'écart type égale à 0,1 et l'intensité lumineuse I est égale à $I_{opt} = \sqrt{K_{sI}K_{iI}}$. De plus, afin de tester la robustesse des algorithmes de commande par rapport aux incertitudes du modèle, les paramètres du système réel sont incertains de 30% au maximum.
Les figures 1.7 et 1.8 présentent l'évolution de la concentration en biomasse et du taux de dilution. Les résultats obtenus montrent que la diminution du taux de dilution entraine une augmentation de la concentration en biomasse et vice versa. D'autre part, on peut remarquer aussi que les lois de commande (r)RNMPC et LRMPC ont de meilleures performances que la commande prédictive NMPC en termes de précision de poursuite. Dans le cas de la NMPC, la présence d'une grande erreur de poursuite est due au fait que la dispersion entre le modèle et le système n'est pas prise en compte dans l'étape de prédiction pendant l'optimisation. Par ailleurs, la RNMPC régule mieux la concentration en biomasse que la rRNMPC et LRMPC mais induit une grande charge de calcul comme indiqué dans le tableau 1.3. En revanche, la méthode rRNMPC permet de réduire le temps de calcul mais on constate une nette diminution de la qualité de poursuite. Enfin, la stratégie LRMPC permet d'avoir le meilleur compromis entre la précision de poursuite et le temps de calcul.

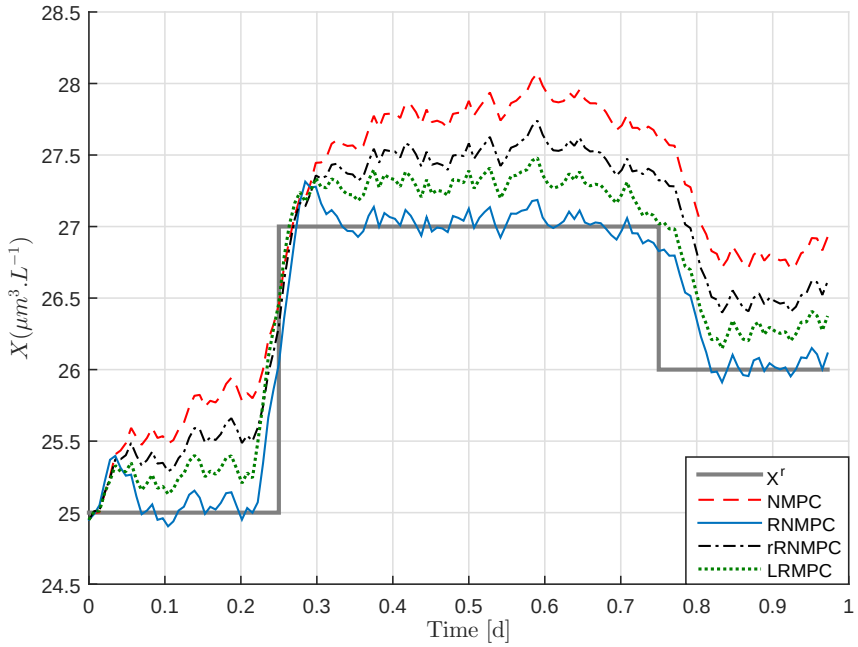

FIGURE 1.7 – Évolution temporelle de la concentration en biomasse.

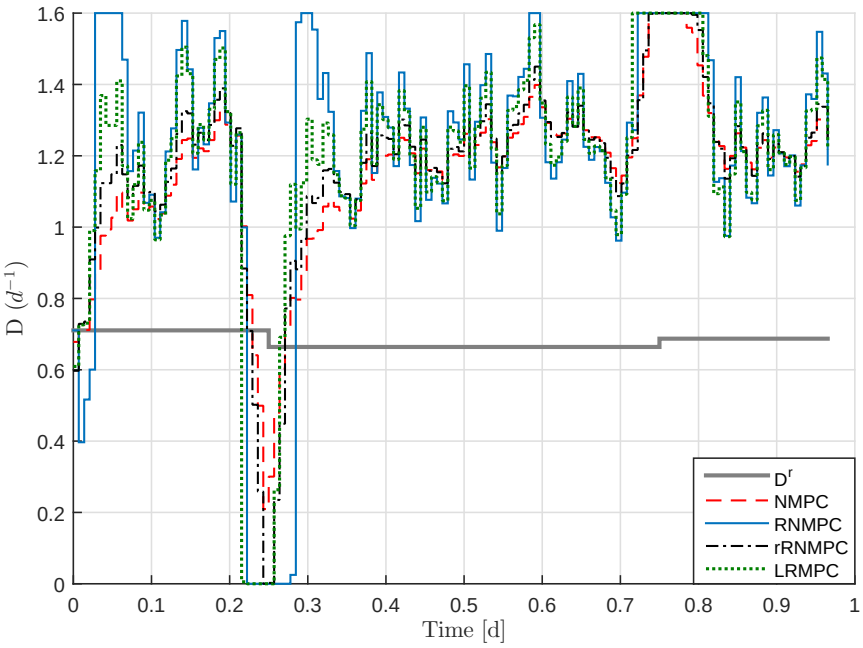

FIGURE 1.8 – Évolution temporelle du taux de dilution.

Tableau 1.3 – Comparaison des algorithmes prédictifs en termes de temps de calcul à chaque instant d'échantillonnage.

	Temps de calcul (s)		
Ind. perf. / Algo.	min	moy	max
NMPC	$< 10^{-5}$	0,014	0,29
RNMPC	0,55	1,85	19,31
rRNMPC	0,047	0,078	0,36
LRMPC	$< 10^{-5}$	0,01	0,09

Dans un deuxième temps, trois lois de commande sont comparées : NMPC, LRMPC et la stratégie de commande hiérarchisée (LRMPC-PI). À l'aide d'une approche de type Monte Carlo, une analyse statistique de la robustesse est effectuée. 100 tests ont été réalisés en désadaptant les paramètres de manière aléatoire et non corrélée entre -30% et +30% (figure 1.9). Les résultats de ces tests sont illustrés figure 1.10 par un histogramme présentant la répartition de la moyenne de l'erreur de poursuite. Les conditions opératoires choisies sont identiques à celles présentées au paragraphe précédent.

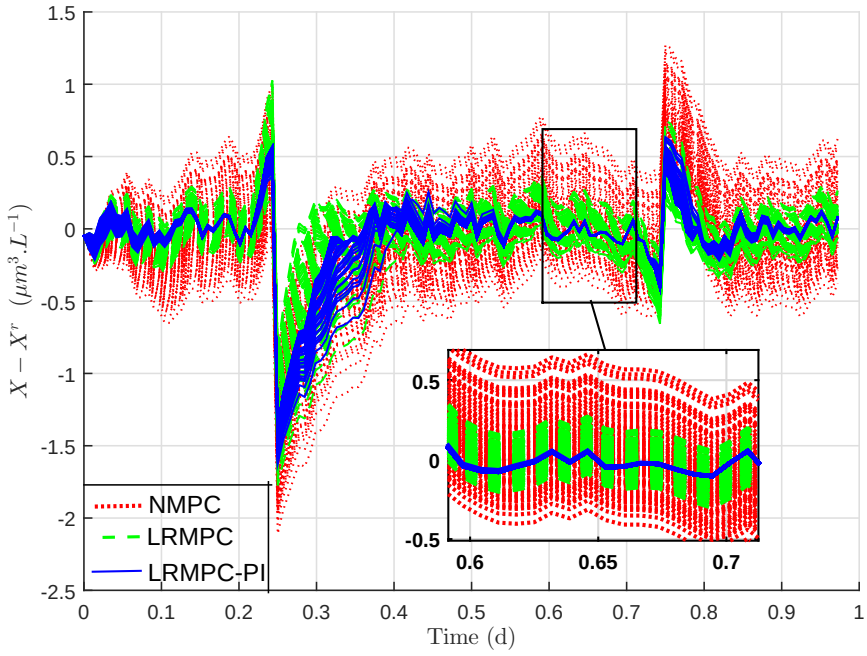

FIGURE 1.9 – Évolution temporelle de l'erreur de poursuite pour une variation aléatoire non-corrélée de tous les paramètres (Monte-Carlo).

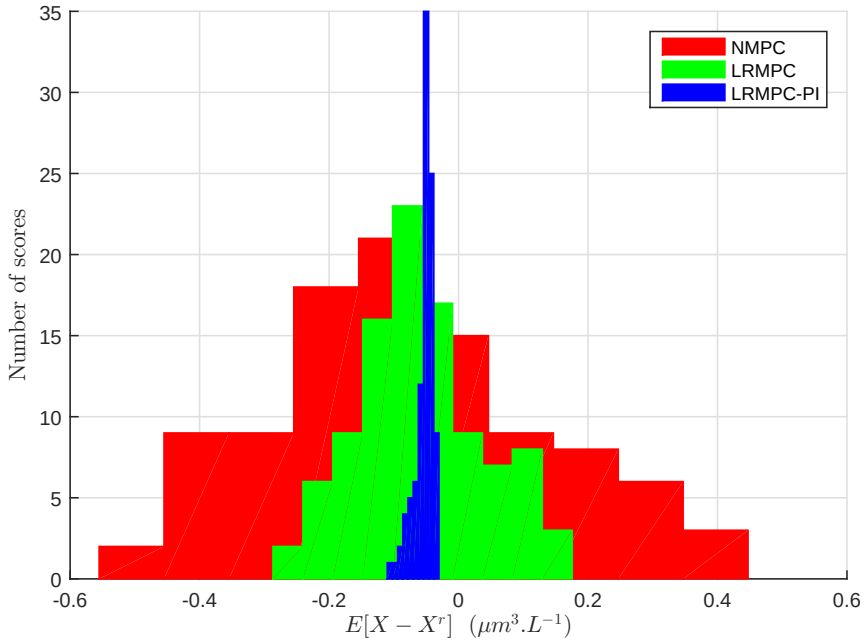

FIGURE 1.10 – Histogramme de la moyenne de l'erreur de poursuite pour une variation aléatoire non-corrélée de tous les paramètres (Monte-Carlo).

On constate que la stratégie hiérarchisée LRMPC-PI permet d'avoir la poursuite la plus précise en termes d'écart type sur l'erreur de poursuite comme indiqué dans le tableau 1.4.

Tableau 1.4 – Comparaison des lois de commande en termes de distribution de l'erreur de poursuite.

Ind. perf. / Algo.	moyenne	écart type
NMPC	-0,07	0,219
LRMPC	-0,06	0,098
LRMPC-PI	-0,05	0,014

Ainsi, à l'aide de cet outil statistique on peut conclure que la structure hiérarchisée permet d'annuler l'erreur de poursuite due à l'étape de linéarisation dans la partie synthèse prédictive.

En dernier lieu, on compare la commande prédictive robuste linéarisée (LRMPC) avec deux stratégies hiérarchisées (LRMPC-PI et LRMPC-ISM) en absence de bruit de mesure. L'objectif de commande est de maintenir la concentration en biomasse à une valeur de consigne fixe ($X^r = 25\mu\text{m}^3/L$) en dépit des perturbations sur l'intensité lumineuse (figure 1.13).

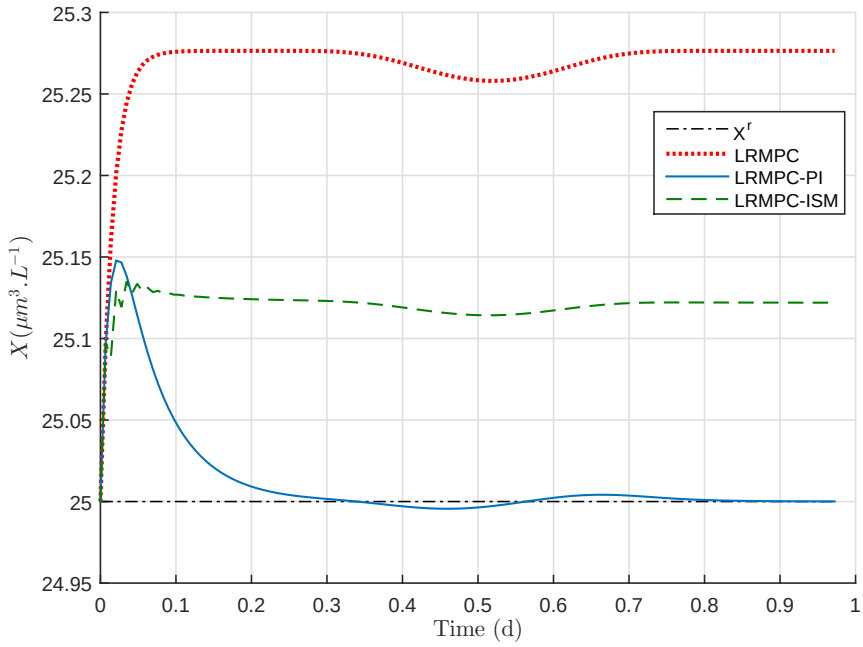

FIGURE 1.11 – Évolution temporelle de la concentration en biomasse.

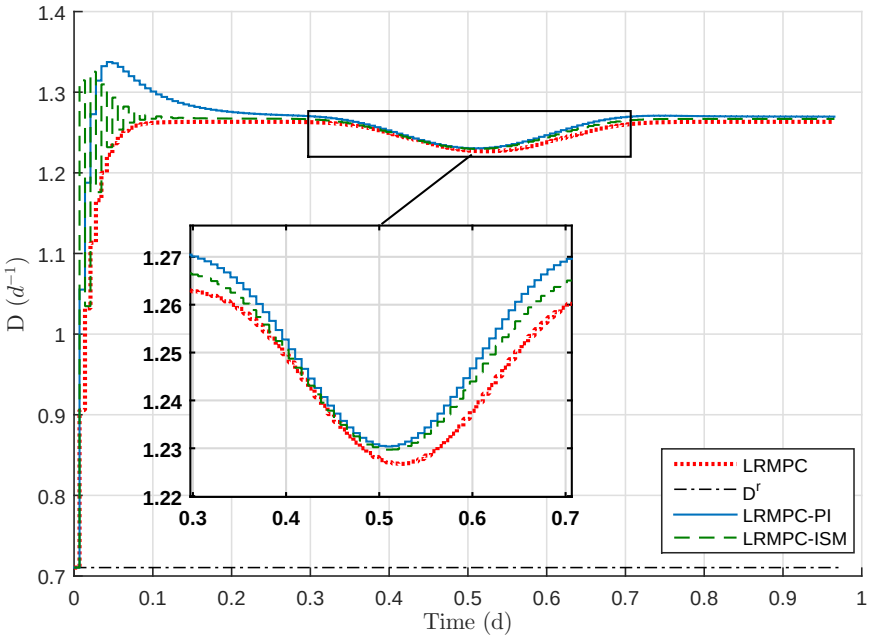

FIGURE 1.12 – Évolution temporelle du taux de dilution.

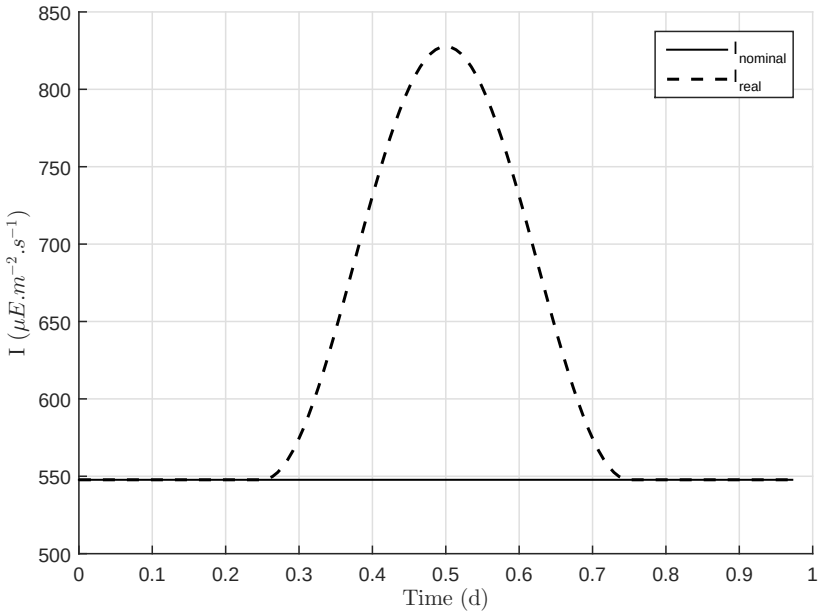

FIGURE 1.13 – Perturbation sur l'intensité lumineuse.

La figure 1.11 montre que les trois lois de commande arrivent à maintenir la concentration en biomasse constante en présence de perturbations sur l'intensité lumineuse. Dans le cas de la LRMPC, on constate la présence d'une importante erreur statique due à l'étape de linéarisation durant la prédiction. En revanche, la stratégie LRMPC-PI arrive à maintenir la sortie autour de la valeur de consigne grâce à l'action du régulateur PI. En ce qui concerne la seconde commande hiérarchisée (LRMPC-ISM), on note la présence d'une certaine erreur statique non négligeable due à l'évaluation du taux de croissance aux valeurs nominales des paramètres dans l'expression de la loi de commande par mode glissant. En effet, ce dernier point constitue l'inconvénient le plus important de l'utilisation de la commande par mode glissant.

1.6 Conclusions et perspectives

1.6.1 Conclusions

Dans cette thèse, des algorithmes de commande prédictive robustes pour des systèmes non linéaires à temps discret soumis à des incertitudes paramétriques ont été proposés. Les principales contributions de cette thèse peuvent être résumées comme suit :

- La première partie propose la mise en œuvre d'une stratégie de com-

mande prédictive non linéaire permettant la poursuite d'une trajectoire de référence pour un système non linéaire à temps discret.

- Dans la deuxième partie, la commande prédictive robuste vis-à-vis des incertitudes paramétriques est formulée en un problème d'optimisation de type min-max. Afin de réduire la charge de calcul induite par la formulation min-max, l'approche proposée consiste à simplifier le problème d'optimisation initial, via l'analyse de sensibilité du modèle vis-à-vis de ses paramètres en linéarisant les sorties prédites autour des valeurs nominales des paramètres du modèle et de la séquence de commande de référence. Deux méthodes d'implantation ont été développées. Dans la première méthode ne tenant pas compte des contraintes, la séquence de commande optimale est calculée à partir d'un problème d'optimisation scalaire. La seconde méthode permet de prendre en compte explicitement les contraintes inégalité sur l'entrée de commande.

- Deux améliorations de la loi de commande prédictive robuste linéarisée ont été proposées dans la troisième partie. Tout d'abord, la linéarisation des sorties prédites autour des valeurs nominales des paramètres du modèle et de la séquence de commande optimale obtenue à l'itération précédente. Cette modification permet de rendre la solution moins sensible aux bruits de mesure. Ensuite, une structure de commande hiérarchisée combinant la commande prédictive robuste linéarisée et une commande auxiliaire (proportionnel intégrateur ou mode glissant intégrateur) est développée afin d'éliminer toute erreur statique en suivi de trajectoire de référence.

- Le dernier chapitre est dédié à l'application des lois de commande développées à un cas d'étude de commande de photobioréacteur de culture de microalgues pour la régulation de la concentration en biomasse en présence d'incertitudes paramétriques, bruit de mesure et fluctuations de l'intensité lumineuse.

1.6.2 Perspectives

Dans la continuité de ces travaux, les perspectives suivantes peuvent être envisagées :

- Application des lois de commande proposées au cas d'un système multivariable comprenant un nombre de paramètres incertains plus important.

- Extension de la formulation du problème d'optimisation aux cas des perturbations et des cinétiques incertaines.
- Synthèse d'un observateur afin d'estimer le vecteur d'état à partir des mesures.
- Amélioration de la qualité du modèle linéarisé en utilisant un développement en série de Taylor limité au second ordre au lieu du premier ordre.
- Étude de l'influence de la résolution du problème d'optimisation en utilisant la dualité lagrangienne sur la stabilité du système en boucle fermée.
- Analyse de la stabilité du système bouclé avec la stratégie de commande hiérarchisée.
- Détermination des conditions à satisfaire pour assurer la stabilité du système avec le couplage régulateur/estimateur.

Chapter 2

Introduction

2.1 Context and motivations

The control of nonlinear systems subject to physical constraints on the input and state is undoubtly a challenging and important problem. The well known systematic nonlinear control methods such as feedback linearization and constructive Lyapunov-based methods lead to very elegant solutions, but they depend on a complicated design procedure that does not scale well to large systems and they cannot handle constraints easily or in a systematic manner. Based on this, the concept of optimal control and in particular the Model Predictive Control (MPC) approach and its nonlinear version NMPC appears to be an attractive alternative since the complexity of the control design increases moderately with the size and complexity of the system. The NMPC strategy is therefore put forward because of its ability to deal with uncertainties and constraints. The essence of NMPC is to optimize, over an open-loop time sequence of controls, the process response using a model of the system to forecast the future process behavior over a prediction horizon.

Moreover, even if NMPC strategies have proved to be efficient in many industrial applications due to their ability to operate the process safely under physical constraints, the monitoring of complex, highly nonlinear, uncertain systems becomes more and more a delicate task. This is the case for example of bioprocesses, that will be the application field considered in this work. Indeed, in such cases, model parameters are generally known with a confidence interval only (determined from an identification procedure for example). The main objective is thus to elaborate an adequate robust control strategy in order to guarantee that the process will yield the reference trajectory under model parameter uncertainties.

Therefore, NMPC strategies must be extended to provide robustness fea-

tures, developing robust control strategies, which can compensate for the lack of parameters information and/or accuracy. There are two popular alternatives for making decision with incomplete knowledge: the stochastic solution (probabilistic theory) and the min-max solution (game theory), which appear as expensive approaches.

Indeed, the robust NMPC (RNMPC) law can be formulated as a nonlinear min-max optimization problem where the effect of the uncertainties must be taken into account in the design procedure. However, this approach tends to become too complex to be solved numerically online. Consequently, the total calculation time is an important factor that must be reduced as much as possible.

As a consequence, the challenge that must be taken up in this work is to design a MPC controller that would be robust against unknown but bounded model parameters uncertainties and computationally more tractable in calculating the optimal control, which makes it suitable for online implementation. For that purpose, starting from the standard RNMPC formulations, new structures will be developed to match this compromise between robustness and computational load, with a dedicated application to bioprocesses.

2.2 Outline

We briefly summarize the main results of this thesis, chapter by chapter, as the following:

- **Chapter 3:** In this chapter, a state of the art of existing Model Predictive Control (MPC) techniques is given with a focus on nonlinear control problems. It starts with the basic MPC problem formulation and provides a review of linear and nonlinear MPC schemes, addresses robustness issues and presents briefly nominal and robust stability concepts.

- **Chapter 4:** This chapter investigates the Nonlinear MPC formulation in order to solve a trajectory tracking problem for continuous/discrete-time nonlinear systems. A continuous/discrete formulation is considered in this thesis, since models of bioprocesses are time-continuous but with discrete-time measurements. A variant of NMPC is presented in section 4.3. It consists in considering the accumulation of the gap between the system and the model at each sampling time over prediction intervals. An illustrative example (for a bioprocess) is presented in section 4.4 and numerical results are provided in order to demonstrate the

effectiveness of the NMPC strategy and highlight its drawbacks and limitations, which motivate the development of RNMPC strategies.

- **Chapter 5:** This chapter deals with a robust formulation of the NMPC in order to take into account the model parameters mismatch. The robust NMPC (RNMPC) law is formulated as a nonlinear min-max optimization problem. However, this approach tends to become too complex to be solved numerically online. To overcome this problem, two major solutions have been proposed. The first one consists in reducing the number of uncertain parameters through a sensitivity analysis. Only the main influencial parameters on the model are considered in the min-max problem whereas the other parameters are kept constant and equal to their nominal values. Secondly, a new approach has been proposed, based on the linearization of the predicted trajectory to turn the original optimization problem into a more tractable one. It consists in simplifying the min-max optimization problem through a sensitivity analysis of the model with respect to its parameters, in order to reduce the calculation time, which makes it suitable for online implementation. The criterion is linearized around the model parameters nominal values and the reference control sequence (control inputs over the prediction horizon). The derived optimization problem is an unidimensionnal optimization problem (referred to as LRMPC). Moreover, stability analysis of the closed-loop system while applying the LRMPC law is established under some assumptions. Furthermore, the inequality constraints on the optimization variables are taken into account in the LRMPC problem formulation. This leads to a bilevel problem (referred to as CLRMPC), with a scalar optimization problem in the upper level, and a quadratic programming one in the lower level. The performances of the proposed control strategies will be illustrated in simulations (the same system as in chapter 4).

- **Chapter 6:** This chapter proposes two improvements for the (C)LRMPC law. The first one consists in linearizing the model at each sampling time along the nominal trajectory which is defined based on the nominal parameter values and the optimal control sequence obtained at the previous iteration, instead of reference control sequence as done in the previous chapter. This modification is motivated by the fact that the optimal control sequence obtained at the previous iteration is non-model-based. As a consequence, the proposed approach appears to be less sensitive against measurement noise thanks to the penality term on the control evolution. The derived optimization prob-

lem is a bilevel one or a scalar optimization problem depending on the considered constraints. However, the proposed control strategies have accuracy issue as they deal with a first-order approximation of the nonlinear system. A hierarchical control strategy is proposed in order to cancel the residual tracking error due to the linearization drawback and possible disturbances. The proposed control strategy combines a robust model predictive control law with a Proportional Integral (PI) law or Integral Sliding Mode (ISM) controller. The predictive controller guarantees the tracking of the reference trajectory, whereas the other regulator ensures a good tracking accuracy. These two improvements are applied to the illustrative example.

- **Chapter 7:** In this chapter, all proposed approaches in the previous chapters are applied to a case study to control of microalgae culture in a continuous photobioreactor. The aim is to regulate the microalgae concentrations at a given reference value, despite disturbances (light intensity fluctuation) and model parameters uncertainties. First, the model of the bioprocess is described and analysed. Secondly, the determination of the equilibrium is addressed. Finally, simulation results are given for reference trajectory tracking and disturbance rejection.

- **Chapter 8:** The last chapter summarizes the developed work in this PhD thesis and proposes some recommendations for future directions.

2.3 List of Publications

The thesis work has resulted in several accepted/submitted publications to international journal and conferences. They are given hereafter:

Journal paper

- S.E. Benattia, S. Tebbani and D. Dumur. Multivariable robust MPC of heterotrophic microalgae fed-batch bioreactor. *To be submitted to Journal of Process Control*, 2016.

Proceedings

- S.E. Benattia, S. Tebbani and D. Dumur. Nonlinear model predictive control for regulation of microalgae culture in a continuous photobioreactor. *Proceedings of the 22nd Mediterranean Conference on Control and Automation*, Palermo, Italy, pp. 469-474, June 16-19, 2014.

- S.E. Benattia, S. Tebbani, D. Dumur and D. Selisteanu. Robust nonlinear model predictive controller based on sensitivity analysis - application to a continuous photobioreactor. *Multi-Conference on Systems and Control*, Antibes/Nice, France, pp. 1705-1710, October 8-10, 2014.

- S.E. Benattia, S. Tebbani and D. Dumur. Robust nonlinear model predictive control of microalgae culture in a continuous photobioreactor. *1st IFAC Conference on Modelling identification and Control of Nonlinear Systems*, Saint-Petersburg, Russia , pp. 192-197, June 24-26, 2015.

- S.E. Benattia, S. Tebbani and D. Dumur. Hierarchical control strategy based on robust MPC and integral sliding mode - application to a continuous photobioreactor. *5th IFAC Conference on Nonlinear Model Predictive Control*, Seville, Spain, pp. 212-217, September 17-20, 2015.

- S.E. Benattia, S. Tebbani and D. Dumur. Hierarchical predictive control strategy of microalgae culture in a photobioreactor. *19th International Conference on System Theory, Control and Computing*, Cheile Gradistei-Fundata, Romania , pp. 231-236, October 14-16, 2015.

- S.E. Benattia, S. Tebbani and D. Dumur. Robust model predictive control - application to microalgae cultivation. *Submitted to Conference on Decision and Control*, Las Vegas, USA, 2016.

Conferences with abstract

- S.E. Benattia, S. Tebbani, D. Dumur. Nonlinear model predictive control of microalgae culture. *Young Algaeneers Symposium*, Montpellier & Narbonne, France, April 3-5, 2014.

- S.E. Benattia, S. Tebbani, D. Dumur. Regulation of microalgae culture in a continuous photobioreactor. *ESBES-IFIBIOP Symposium*, Lille, France, September 7-10, 2014.

Others (***oral presentation***)

- S.E. Benattia, S. Tebbani and D. Dumur. Robust nonlinear model predictive control for regulation of microalgae culture in a continuous photobioreactor. *Journée du Groupe de Travail de Commande Prédictive Non Linéaire*, Paris, June 11, 2015.

- S.E. Benattia, S. Tebbani and D. Dumur. Hierarchical robust predictive control strategy of microalgae cultivation process. *Journée du Groupe de Travail de Commande Prédictive Non Linéaire*, Paris, March 24, 2016.

Chapter 3

Predictive control: State of the art and main strategies

3.1 Introduction

The purpose of this chapter is to give an introduction to one of the most powerful strategies that can be used to address nonlinear control problems. We present the basic MPC problem formulation and provide a review on previous works that appear in the literature. The key advantages/drawbacks of MPC are outlined and some of the theoretical, computational and implementation aspects are discussed. It also serves as a centralized literature study for the following chapters. In any case, this introduction does not represent an exhaustive overview of predictive control, it however provides the reader with the sufficient background for the next chapters.

3.2 Background

The nonlinear optimal control theory was developed in the 1950's and 1960's, resulting from the maximum principle of Pontryagin [116] and dynamic programming method developed by Bellman [12]. The receding horizon principle was proposed by Propoi [117], within the frame of open loop optimal feedback. The philosophy of the receding horizon problem is closely related to the minimum time optimal control problem and to linear programming [154]. In the late 1970s, due to the progress in algorithms for solving constrained linear and quadratic optimization problems, linear MPC, also referred to as receding or moving horizon control, became popular in control engineering. As a consequence, the interest in MPC for process control grew quickly. Many works appeared principally Richalet *et al.* [129] and Cutler *et al.* [40] were

the first in the control community to propose industrial applications of MPC strategy. MPC has become the de-facto standard advanced control method in the process industries due to its ability to handle large scale multivariable processes. Many systems are however in general inherently nonlinear. In this case, linear models are often inadequate to describe the process dynamics and nonlinear models have to be used.
The practical interest is driven by the fact that process nonlinearities and constraints are explicitly considered. This motivates the use of nonlinear model predictive control (henceforth abbreviated as NMPC), the extension of well established linear MPC. MPC involves the repetitive solution of an optimal control problem at each sampling instant in a receding horizon window. The key challenge that has to be met is to ensure that the implementation of a sequence of open-loop optimal control values will be stable, when considering the closed loop system. Subsequently, NMPC problems have witnessed a steadily increasing attention from academic control theoretists. This leads to increasing advances with focus on design that guarantee stability and robustness. A lot of pioneer works have emerged in this topic: Chen and Shaw [31], Keerthi and Gilbert [77], Michalska and Mayne [102], Parisini and Zoppoli [113], Alamir and Bornard [2], Magni and Sepulchre [95], Chen and Allgöwer [32], De Nicolao *et al.* [43], Mayne *et al.* [100], and Jadbabaie and Hauser [74]. In addition to the research on NMPC, researchers have developed so-called Real Time Optimization (RTO) [30] approaches which are similar to NMPC, as they are generally based on nonlinear programming, and use static nonlinear models, while NMPC uses dynamic nonlinear models.

Receding Horizon Control (RHC) is cited as one of the most popular advanced techniques for industrial process applications and has been widely adopted in the field of process control, due to the simplicity of the algorithm. MPC has induced a significant impact on industrial control engineering world, as a consequence, there are many applications of predictive control strategy in very various domains, *e.g.* robot manipulators [83], clinical anesthesia [89], cement industry and pulp factories [132], drying towers and robot arms [35], distillation columns [70, 126], Polyvinyl chloride (PVC) plants, steam generators and oils refining [129], solar power plant [27], thermomechanical pulping [66, 108], biochemical systems [7, 143, 141], motor control and food extruder process [149], petrochemical sector [58], aerospace [67, 21, 39], automotive [124], mining metallurgy [8, 109], *etc.*

MPC is a powerful approach of great promise that has proven itself in several applications for many classes of systems. On the other side, nonlinear MPC is limited in its industrial impact due to the challenges of guaranteeing a global solution to the resulting nonlinear optimization problem within the real time requirements. For additional information about the use of linear

and nonlinear MPC in practice, applications can be found in many articles and books, *e.g.* in [65, 3, 79]. It is not so easy to list all referenced MPC applications, more can be found in [118].

3.3 The basic principles of MPC

We consider a general discrete-time, state-space nonlinear model of the plant, in the form

$$x_{k+1} = f(x_k, u_k) \tag{3.1}$$

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are the plant state and control action vectors at time step k, respectively.
Given the plant (3.1), a general formulation of MPC problem can be described as the following

$$\min_{u_k,\ldots,u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} \phi(\hat{x}_{k+i}, u_{k+i-1}) \tag{3.2}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k \end{cases} \tag{3.3}$$

where

- N_p is the prediction horizon.
- $\hat{x}_{k+i}$ is the predicted state value.
- x_k is the initial condition which is the plant state at time step k.
- $\phi(.,.)$ is the cost per stage that is generally a quadratic term that minimizes the difference between the predicted state and the setpoint. However, other norms can be considered such as the L_1 or infinity norms.

The MPC controller is implemented in a moving horizon framework. At current time step k, the plant state x_k is used as the initial condition and the optimization problem (3.2)-(3.3) is solved over a horizon of length N_p. However, only the first calculated control action is implemented, *i.e.* u_k. At time step $k+1$, we move the time frame one step ahead and the problem (3.2)-(3.3) is solved with the new plant state x_{k+1} as the initial condition according to the receding horizon principle.
The overall basic structure of an MPC control loop is depicted in Figure 3.1.

All the MPC algorithms have the following common elements.

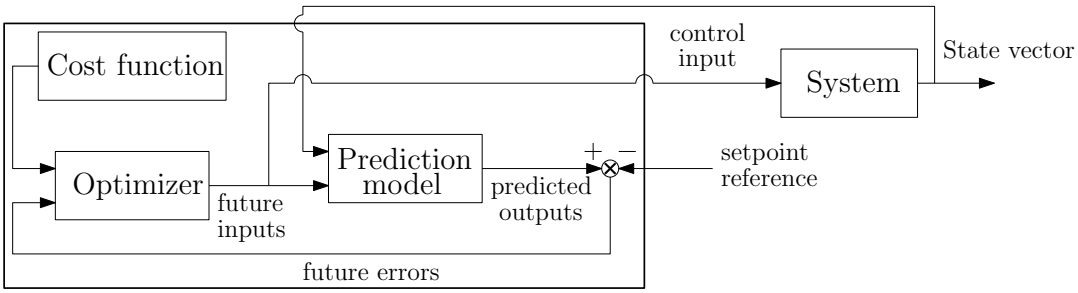

Figure 3.1: Structure of the Model Predictive Control strategy.

3.3.1 Prediction model

A system model is a mathematical representation of the reality. It aims at predicting the time evolution of each state variable. Generally, models are related to both parameters, input and output signals. The explicit use of the model (3.3) is determined by the necessity to calculate the predicted states $\hat{x}_{k+i}$ for a determined horizon of length N_p. These predicted states depend on the known values up to time instant k (*i.e.* x_k) and on the future control sequence, u_{k+i}, $i = \overline{0, N_p - 1}$, which are the optimization variables. The process model plays, as a consequence, a decisive role in the controller performance. Moreover, the design of the model should contain the necessary mechanisms for obtaining the best possible model while being simple to be implemented and to be understood. Various models can be considered, the following being the most commonly used [27]: impulse response, step response, transfer function, state space, neural networks and fuzzy models.

3.3.2 Cost function

For setpoint tracking MPC, the index to be optimized is a quadratic (non-negative) function measuring the distance between the predicted model state $\hat{x}$ and a determined known reference sequence x^r over the horizon window, plus a quadratic function weighting the control effort necessary to achieve the specified goal.
If the quadratic norm is used (which is usually the case), the general expression for such an objective function can be:

$$\sum_{i=1}^{N_p} \phi(\hat{x}_{k+i}, u_{k+i-1}) = \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + \sum_{i=1}^{N_c} ||\Delta u_{k+i-1}||^2_{\beta_i} \tag{3.4}$$

with $\Delta u_{k+j} = 0, \quad j = \overline{N_c, N_p - 1}$ and $\Delta u_{k+i-1} = u_{k+i-1} - u_{k+i-2}$.
The criterion usually considers the following tuning parameters:

- N_p and N_c are the length of the prediction horizon and the control horizon respectively, which are not necessarily equal (*i.e.* $N_p \geq N_c \geq 1$) in order to reduce the computational load.
- x^r denotes the reference state trajectory which is used to specify closed-loop behavior and tracking performances.
- $\alpha_i \in \mathbb{R}^{n_x \times n_x}$ and $\beta_i \in \mathbb{R}^{n_u \times n_u}$ are the weighting matrices for the state tracking error and the control, respectively. The form of the cost function in (3.4) implies that the deviations of the predicted controlled states $\hat{x}$ from a reference x^r and the control moves are penalized at every point in the prediction horizon.

A conceptual picture of the prediction horizon is shown in Figure 3.2.

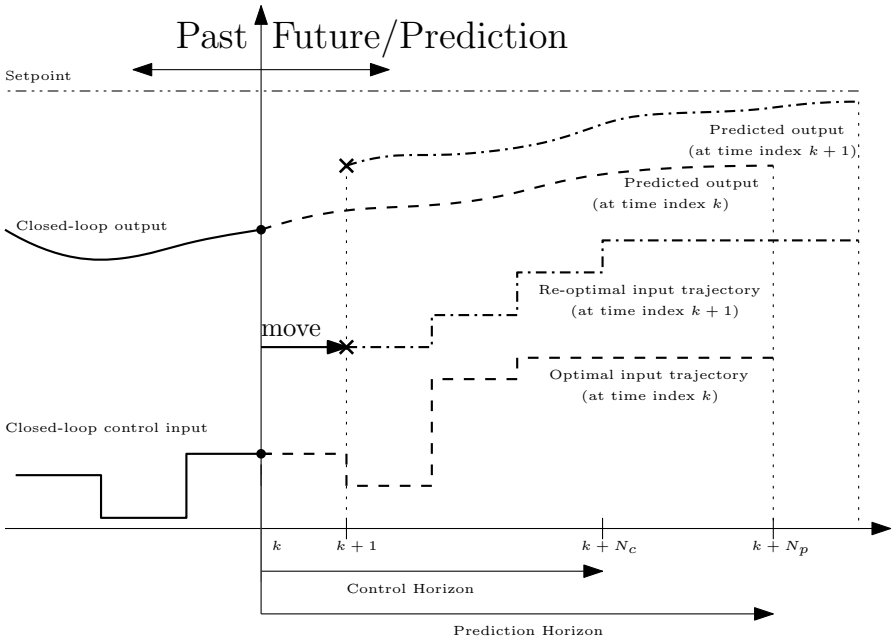

Figure 3.2: Receding horizon principle: the basic idea (SISO case).

These tuning parameters should be adjusted to avoid unstability and to give satisfactory dynamic performance, since all affect the behavior of the closed-loop combination of system and predictive controller. It should be noted that it is possible to use a non-quadratic cost, so that *e.g.* absolute values of errors can be penalized, rather than square values as detailed in [90].

In addition, constraints can be taken into account in the optimization problem. For example, the actuators may have a limited field of action. By adding these constraints to the optimization problem, the latter becomes more complex, so that the solution can not be obtained explicitly as in the unconstrained case. Moreover, in this case, the paradigm of finding a feasible solution to the optimization problem has to be addressed.

3.3.3 Control law calculation

The optimizer is a fundamental part of the MPC strategy as it provides the control actions by minimizing the cost function (3.4). An analytic solution can be obtained for the quadratic criterion case if the model is linear and there are no constraints. Otherwise, the control sequence results from a real-time optimization strategy, involving solving a nonlinear programming problem (NLP) (for more details see section 3.5). The size of the optimization problem depends on the number of variables and on the prediction horizon considered.
Summarizing, the basic MPC scheme works as follows:

1. Obtain measurements
2. Compute the model prediction over a certain prediction horizon
3. Get an optimal input signal by minimizing a given cost function
4. Implement the first value of the optimal input signal until new measurements are available.
5. Continue with 1.

While it can be desirable for computational and performances reasons to choose unequal lengths of the prediction and control horizon, we assume in the following that $N_p = N_c$ for the study.

3.4 Literature review

In general, MPC can be split into two different classes of methodologies: linear and nonlinear. Maciejowski and Camacho provide a good overview of linear theory [90, 27], while Allgöwer and Zheng provide an overview of nonlinear methods [3]. The various MPC algorithms result from the way the cost function to be minimized and the constraints of the system are specified. This section aims at briefly presenting the main MPC strategies found in the

literature, for each of them the reader is referred to appropriate references for more details.

3.4.1 Linear case

Generally, linear MPC uses a low order linear model to represent the system. In case of linear constraints, it has the advantage of solving small optimization problem, which can be done fast enough at each sampling time in order to be implemented in the moving horizon framework. In this case, the solution of the optimization problem is the solution of a quadratic or linear program (QP/LP), which is known to be convex and for which there exists a variety of numerical methods and softwares.

***Remark* 3.1.** An analytic solution can be obtained for the quadratic criterion if the model is linear and there are no constraints.

On the other hand, linear MPC also suffers from drawbacks such as plant-model mismatch because the model may be only valid around an operating point. Depending on different modelling and solution strategies, there are many variations of linear MPC, methods considered to be the most representative will be briefly cited [27]:

- **Dynamic Matrix Control (DMC)** [40]: the DMC uses a step response model of the process. The output of the plant is then expressed through the time response parameters. The setpoint trajectory can be approached by a smoother trajectory (called projected desire trajectory). Shell engineers developed QDMC by posing the DMC problem as a quadratic programming (QP) problem in which constraints appear explicitly. Formulations based on linear impulse response models can be found in [110].

- **Model Algorithmic Control (MAC)** [128, 129, 121]: this approach is also known as model predictive heuristic control. It uses an impulse response model (only for stable processes) and introduces a reference trajectory as the output of a first order system which evolves from the actual output to the setpoint according to a determined time constant.

- **Predictive Functional Control (PFC)** [127, 125]: this strategy is dedicated to fast processes although it can accomodate all kinds of systems and uses a state space model representation of the process. It has two distinctive characteristics: the use of coincidence points by considering only a subset of points in the prediction horizon and the

parametrization of the control signal by means of a set of polynomial basis functions.

- **Extended Prediction Self Adaptive Control (EPSAC)** [41]: in this method, the process is modelled by a discrete transfer function for prediction. The control signal is set constant over the prediction horizon. The feedforward effect can be included by extending the model with a measurable disturbance term. The control signal can be calculated analytically leading to a very simple control law structure.

- **Extended Horizon Adaptive Control (EHAC)** [153]: the implementation of EHAC is similar to the previous method but it allows a longer time to drive the plant output to its reference value. For prediction, the process is modelled by a transfer function without taking a model of the disturbances into account. It aims at minimizing the discrepancy between the model and the reference at final instant of the prediction window.

- **Generalized Predictive Control (GPC)** [36]: the GPC calculates a control sequence in a such way that it results from the minimization of a multi-stage cost function defined over a prediction horizon. An analytic solution is derived in the unconstrained case. Unstable and non-minimum phase plants could be dealt with.

3.4.2 Nonlinear case

There are some processes for which the nonlinearities are so crucial or so important that a linear model is not sufficient. For these systems, a linear MPC will not be very effective, so some solutions have been proposed to cope with this problem. Using a nonlinear model changes the problem from an online convex quadratic problem to a repeated online possibly non-convex nonlinear problem, which is more difficult to solve. As a consequence, the convergence and success of a given optimization problem depend largely on the initial guess provided for the solution. Nonlinear model predictive control (NMPC) is an optimization-based method for the feedback control of nonlinear systems. Its primary applications are stabilization and tracking problems. NMPC is based on the receding horizon principle, where a finite horizon open-loop optimal control problem is solved at each sampling instant and the optimized control trajectory is implemented until a new optimized control trajectory is available at the next sampling instant. Its philosophy is therefore similar to linear MPC. The key characteristics of the NMPC are as follows:

- it allows the use of an explicit nonlinear model for prediction.
- it allows the explicit consideration of state and input constraints.
- specified performance criteria can be minimized online.
- it requires solving online an open-loop optimal control problem.
- system states must be measured or estimated in order to perform the prediction.

***Remark* 3.2.** It should be noted that except the first point, all others are present in the linear MPC law.

This subsection aims at drawing up a picture of some classical NMPC algorithms:

- **Explicit schemes** [13, 64, 112]: they compute the control sets offline by enumerating all the possible states or their approximations. Then, online control actions are chosen from these sets based on where the state lies.
- **Infinite-horizon NMPC** [77]: this problem has only theoretical significance, because an infinite horizon NMPC formulation is that at every sampling instant an infinite dimensional optimization problem must be solved as follows:

$$\min_{u_k,\ldots,u_\infty} \quad \sum_{i=1}^{\infty} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + ||\Delta u_{k+i-1}||^2_{\beta_i} \tag{3.5}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i \geq 0 \\ \hat{x}_k = x_k \end{cases} \tag{3.6}$$

  Moreover, infinite-horizon NMPC has the property that the open-loop predictions are identical to the closed-loop response in nominal application. Ideally, we would like to use infinite-horizon NMPC formulations due to stability properties. Unfortunately, the main problem is that infinite-horizon schemes can often not be applied in practice.

- **Finite-horizon NMPC with terminal equality constraint** [77]: the infinite horizon NMPC can be approximated by a finite horizon formulation with terminal equality constraints as follows:

$$\min_{u_k,\ldots,u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + ||\Delta u_{k+i-1}||^2_{\beta_i} \tag{3.7}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k, \quad \hat{x}_{k+N_p} = x^r_{k+N_p} \end{cases} \tag{3.8}$$

The terminal equality constraint, $\hat{x}_{k+N_p} = x^r_{k+N_p}$, basically ensures that at the end of the finite horizon, the closed-loop system approaches the same steady state as in the infinite horizon.

- **Finite-horizon NMPC with terminal inequality constraint** [105, 137]: in this method, a terminal constraint set at the end of the horizon is introduced in the problem formulation. The finite-horizon NMPC goal is to steer the plant to the constraint set $\mathbb{X}_{N_p}$ in order to guarantee the stability of the closed-loop system. The optimization problem is then given by:

$$\min_{u_k, \ldots, u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + ||\Delta u_{k+i-1}||^2_{\beta_i} \tag{3.9}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k, \quad \hat{x}_{k+N_p} \in \mathbb{X}_{k+N_p} \end{cases} \tag{3.10}$$

- **Finite-horizon NMPC with terminal cost** [20]: the proposed idea to ensure the closed-loop stability is to add a terminal cost function at the end of the finite-horizon as follows:

$$\min_{u_k, \ldots, u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + ||\Delta u_{k+i-1}||^2_{\beta_i} + T(\hat{x}_{k+N_p}) \tag{3.11}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k \end{cases} \tag{3.12}$$

where $T(\hat{x}_{k+N_p})$ is the terminal cost function.

- **Quasi-infinite horizon NMPC (QIH-NMPC)** [32]: this strategy consists in combining both the terminal cost function and terminal inequality constraint into the finite-horizon formulation.

$$\min_{u_k, \ldots, u_{k+N_p-1}} \quad \sum_{i=1}^{N_p} ||\hat{x}_{k+i} - x^r_{k+i}||^2_{\alpha_i} + ||\Delta u_{k+i-1}||^2_{\beta_i} + T(\hat{x}_{k+N_p}) \tag{3.13}$$

$$\text{s.t} \begin{cases} \hat{x}_{k+i+1} = f(\hat{x}_{k+i}, u_{k+i}), \quad \text{for} \quad i = \overline{0, N_p - 1} \\ \hat{x}_k = x_k, \quad \hat{x}_{k+N_p} \in \mathbb{X}_{k+N_p} \end{cases} \tag{3.14}$$

The terminal state penalty term and the terminal inequality constraint have to be chosen in order to guarantee the asymptotic stability of the closed-loop system [32].

***Remark* 3.3.** Thanks to the principle of optimality, we can recover infinite-horizon NMPC from finite-horizon NMPC with an appropriate choice of the terminal penality $T(.)$ and terminal region $\mathbb{X}_{N_p}$ [32].

- **State/Output-Feedback NMPC** [57]: one of the key obstacles of NMPC is that it is inherently a state feedback control scheme using the current state and system model for prediction. Thus, for an application of predictive control in general, the full state information is necessary and must be reconstructed from the available output information. In many applications, however, the system state can not be fully measured, *i.e.* only some outputs are directly available for feedback. Various researchers have addressed the question of output feedback NMPC using observers for state recovery. To achieve non-local stability results of the observer-based output feedback NMPC controller, two possibilities seem to be attractive:

  - Certainty equivalence approach: the state observer is used as the real system state following the "certainty equivalence" principle. The stability of the closed-loop system is established thanks to the separation of the observer error from the state feedback by time scale separation. Another possibility is to use observers for which the speed of convergence of the observation error can be made sufficiently fast and the absolute achieved observation error can be made sufficiently small. Semi-regional stability results for the closed-loop can be established in this case [57].
  - Taking into account the observation error in the NMPC controller by using some bounds on the estimation error. This solution is closely related to the design of robust stabilizing NMPC schemes and typically requires observers that deliver an estimate of the observation error.

- **Economic NMPC (E-NMPC)** [49]: the E-NMPC has essentially the same characteristics of the formulation and implementation of a NMPC, the only difference is that the objective function is a general economically-oriented cost function of the state and manipulated variables instead of a tracking quadratic function. For example, the objective function for an extractive distillation column [133] is a trade-off between the energy cost, reboiler duty, the production revenue and distillate flow rate.

3.4.3 Predictive Control architecture

There are large-scale systems for which control problems to be solved are too complex by using a unique controller. The reason is that these systems to be controlled are often composed by many interacting subsystems with an increasing complexity, *e.g.* water systems, process plants, interconnected power systems, manufacturing systems and traffic networks. They necessitate new ideas for dividing the analysis and synthesis of the overall system into independent or almost independent subproblems, for dealing with the incomplete information about the system. To overcome this difficulty, many control structures (*i.e.* distributed and decentralized) have been developped [92, 34, 136]. In the sequel, three types of implementation will be presented in order to control the system.

- **Centralized control**: in a centralized control architecture, the controller manages all the subsystems. Figure 3.3 is a schematic representation of a centralized MPC architecture (with y^r the reference, y the output, u the control input and x the state vector).

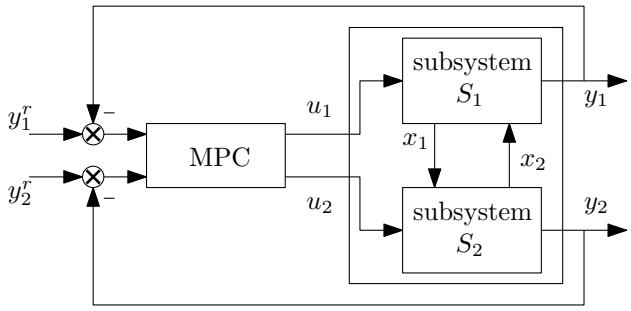

Figure 3.3: Centralized MPC architecture [34].

All the MPC laws presented previously in sections 3.4.1-3.4.2 are centralized control strategies.

- **Decentralized control**: this kind of strategy is based on control architectures where the control input and the controlled variables are gathered into disjoint sets. These sets are then coupled to produce non-overlapping pairs for which local regulators are designed to operate in a completely independent fashion, like the simple example shown in Figure 3.4 (with same notations as in Figure 3.3).
An illustrative application to formation control with collision avoidance for a multi-UAV system can be found in [81].

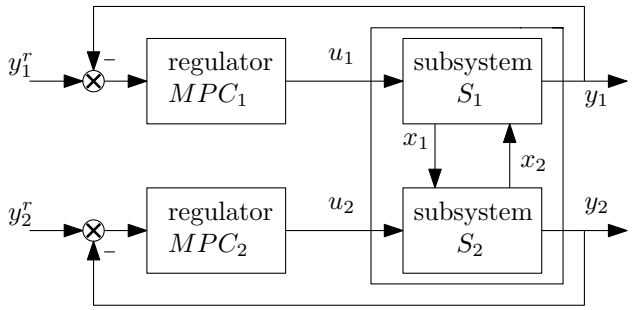

Figure 3.4: Decentralized MPC architecture [34].

- **Distributed control**: in distributed control systems, the achievement of a global control task is obtained by the cooperation of many controllers, each one computing a subset of control commands individualy under a possibly limited exchange of information with the other controllers. In fact, the local regulators are designed with MPC, the information transmitted typically consists in the future predicted control computed locally, so that any regulator can predict the interaction effects over the considered prediction horizon. An example of distributed strategy is reported in Figure 3.5 (with same notations as in Figure 3.3).

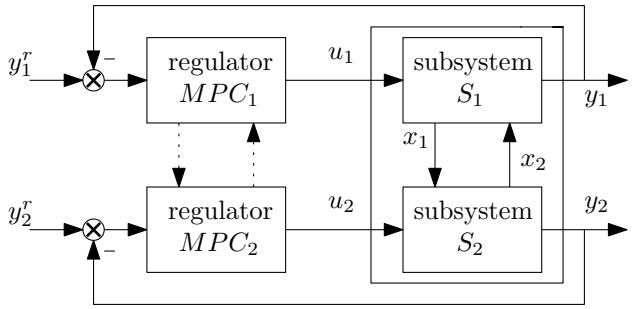

Figure 3.5: Distributed MPC architecture [34].

Several approaches have been proposed in the literature. They differ from each other in the assumptions made on the kind of interactions between different components of the overall system (fully or partially connected algorithms), the model of information exchange between subsystems, and the control design technique used for each subsystem. An illustrative application to Hydro-Power Valleys with French main electricity provider EDF can be found in [155].

3.5 Methods for dynamic optimization

The optimization problems in which the model and constraints are linear and the objective is quadratic are a well studied class of problems, and the dynamic optimization problem is solved using well established QP algorithms. In this section, we discuss solution strategies to efficiently solve nonlinear dynamic optimization problems. In general the solution of a nonlinear (in general non-convex) optimization problem can be computationally expensive. Methods to solve the open-loop optimal control problem (3.2-3.3) can be classified into three different approaches (Figure 3.6):

- **Hamilton-Jacobi-Bellmann partial differential equations/dynamic programming**: this method is based on the direct solution of so-called Hamilton-Jacobi-Bellmann partial differential equations. This approach is valid only for small systems due to the curse of dimensionality, *i.e.* since the complete solution is considered at once, it is in general computationally intractable. Moreover, the inequality constraints on the state variables, as well as dynamical systems with switching points, lead to discontinuous partial derivatives and cannot be easily included. The application of this methodology is restricted to the case of continuous state systems.

- **Euler-Lagrange differential equations/calculus of variations/maximum principle**: classical calculus of variations is used in order to obtain an explicit solution of the input as a function of time.

- **Direct solution using a finite parametrization of the control signal**: The input is parametrized by a finite number of values allowing an approximation of the original open-loop control problem.

The basic idea behind the direct solution using a finite parametrization of the controls is to approximate the original infinite dimensional problem into a finite dimensional nonlinear programming problem. Then, the input control signals are parametrized, *e.g.* using a piecewise constant function over a sampling time. As classified in [18, 51] there are two main strategies:

- **Sequential approach**: in every iteration step of the optimization strategy, the differential equations are solved exactly by a numerical integration, *i.e.* the solution of the system dynamics is implicitly determined during the calculation of the cost function and only the input vector u_{k+i-1} appears directly in the optimization problem while the intermediate states $\hat{x}_{k+i}$ disappear from the problem formulation. This

leads to a Direct Single Shooting strategy (DSS) [80]. This approach is also known as Control Vector Parametrization (CVP) [147];

- **Simultaneous approach**: the simultaneous approach requires not only an initial control trajectory guess, but also one for the state trajectory. The Ordinary Differential Equations (ODE) are discretized in time and the resulting finite set of nonlinear algebraic equations are treated as nonlinear equality constraints. The intermediate states $\hat{x}_{k+i}$ are treated as unkown variables together with the parametrized control signal. This leads generally to a Direct Multiple Shooting strategy (DMS) [46] and Collocation methods [17].

For both approaches, the resulting optimization problem is often solved using sequential quadratic programming algorithm (SQP).

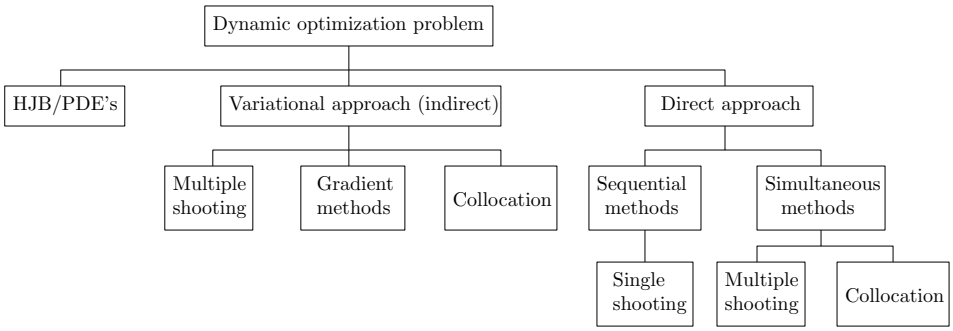

Figure 3.6: Classification of dynamic optimization solution methods [18].

***Remark* 3.4.** For an online solution of the NMPC problem, only the sequential approach is used in general.

In the case of NMPC, some algorithms were proposed to simplify the optimization procedure:

- **Neighboring extremals** [152]: this method was applied to optimal control problems in [25]. It consists in solving a full solution of the optimal control problem offline using indirect methods and perform a full iteration online to find the approximated optimal solution. Moreover, a neighboring extremal update approach was proposed in [152], and more specifically applied to NMPC design. It consists in solving an optimal control problem over a long time horizon. Then, during each sampling time, a fast update, determined from a QP problem, is performed for the control variable.

- **Newton-type controller** [84, 45, 50, 44]: it performs a single full Newton step of NMPC online to allow a quick return of control action. The main advantage is that it rejects the fast disturbance quickly. The Newton's method is based on linearization and it is used in nonlinear programming to solve algebraic equations closely related to the first order optimality conditions of the optimization problem, known as the Karush-Kuhn-Tucker (KKT) conditions. Obviously, the KKT conditions also involve inequalities which means that Newton's method cannot be applied directly.

- **NLP sensitivity-based controller** [26, 75]: it aims at overcoming the drawbacks posed by the Newton-type controllers. NLP sensitivity-based controllers use a direct method to solve the optimal control problem. The sensitivity analysis for NLP provides information on regularity and curvature conditions at KKT points, evaluates which variables play dominant roles in the optimization, and provides first order estimates for parametric nonlinear programs. In closed-loop, this approach is very effective to handle uncertainty while requiring only a minimum number of full real-time optimizations, reducing the online computational load.

- **Advanced step NMPC (asNMPC)** [156]: a kind of NLP sensitivity-based NMPC which uses the current control action to predict the future plant state in order to solve the future optimization problem in advance, while the current sampling period evolves.

3.6 Robust NMPC schemes

Nonlinear MPC is often used in control problems. However, assumptions of nominality reduce its use or performances. This is why other kinds of predictive control methods are developed to take account of uncertainties related to the system model and disturbances acting on it. Hereafter, the main robust NMPC strategies will be briefly presented. There exists a wide range of NMPC formulations that include robustness into the formulation of the optimization problem. We distinguish four main types of approaches:

- **Linear parameter varying formulations**: these methods are based on the reformulation of nonlinear models as linear parameter-varying (LPV) models which allow the use of linear and bilinear matrix inequality (LMI and BMI) formulations [6, 29, 148].

- **Min-max NMPC**: in this formulation, the standard NMPC setup is kept, however, now, the cost function to be optimized considers the worst disturbance sequence occuring. The disturbance/uncertainty in the system is considered as a player working against the input. Thus, the problem is formulated as a classical min-max problem over a finite-horizon. There are two formulations of min-max NMPC:

  - Open-loop formulation: this approach guarantees the robust stability of the system and the robust feasibility of the optimization problem, but it may be conservative since the control sequence has to ensure constraints fulfilment for all possible uncertainty scenarii without considering the fact that future measurements of the state contain information about past uncertainty values [85, 93, 102]. This method may have a small feasible set and a poor performance due to the fact that it does not include the effect of feedback provided by the receding horizon framework.
  - Closed-loop formulation: the min-max problem represents a differential game where the controller is the minimizing player and the disturbance is the output of the maximizing player [88, 91, 93]. The controller chooses the control input as a function of the current state so as to ensure that the effect of the disturbance on the system output is sufficiently small for any choice made by the maximizing player. This method would guarantee a larger feasible set and a higher level of performances in comparison with the open-loop min-max NMPC. By doing this, the reaction of the controller to the uncertainty is incorporated in the prediction and the conservativeness is mitigated.

  The robust min-max strategy and more specifically the closed-loop formulation, will be detailed in Chapter 5.

- H_∞**-NMPC** : the standard H_∞ problem is implemented in a receding horizon framework by considering a particular choice of the cost function. More details concerning this approach can be found in [60, 150, 151].

- **Stochastic Model Predictive Control (SMPC)**: all previous strategies consider a deterministic framework by assuming that pertubations and uncertainties belong to compact bounded sets. However, it can be sometimes more realistic to solve problems based on a probabilistic description of the uncertainties and perturbations. This corresponds to

a stochastic approach. Thus, formulations based on a probabilistic description of uncertainty can also be characterized as problems involving random model uncertainty with known probability distribution. Furthermore, stochastic NMPC can also be characterized similarly to min-max NMPC as described above, either open-loop formulation [28] or closed-loop formulation [63].

3.7 Stability

One of the key questions in the MPC strategy is certainly the stability of the closed-loop system. After basic theoretical principles of MPC had been clarified, more advanced topics like robust stability under perturbations, performance estimates and efficiency of numerical algorithm were addressed. The stability of MPC for constrained nonlinear systems necessitates the use of Lyapunov stability theory that can be expressed conveniently via so-called comparison functions, which were introduced in nonlinear control theory by Sontag [139].
The stability of the system without disturbances is called nominal stability, while the stability of the system in the presence of disturbances is termed robust stability.

3.7.1 Nominal stability

Nominal stability of closed-loop system is proven if one can find a Lyapunov function for the system. Strategies which guarantee the nominal stability are summarized as follows: infinite horizon NMPC, finite-horizon NMPC with terminal equality constraint, finite-horizon NMPC with terminal inequality constraint, finite-horizon NMPC with terminal cost, quasi-infinite horizon NMPC. Results with guaranteed nominal stability are well-addressed by existing works detailed in [100, 106, 122, 123, 157, 130].

3.7.2 Robust stability

As stated before, the robust control design of NMPC is done by taking into account uncertainties in an explicit manner in order to optimize the objective function for the worst situation of the uncertainties. There are many approaches to analyze the robust stability, such as Input-to-State (ISS) framework, robust stability margin and invariant set theory coupled with ISS-Lyapunov stability framework. Further details and a more comprehensive

treatment of this topic can be found in [94, 93, 115, 119, 85, 88, 4, 104]. Fundamental results on stability for constrained and nonlinear predictive control for state-space models are well summarized and categorized in [87, 120].

3.8 Advantages and drawbacks

The RHC presents a series of advantages over the other existing control strategies, summarized as follows:

- Straightforward formulation, based on well understood concepts,
- Open methodology which allows future extensions,
- Compensation for dead times,
- Applicability to a great variety of systems, including nonlinear systems and time-delayed systems,
- Proof of stability for linear and nonlinear systems with input and state constraints, under certain specific conditions,
- Very useful when future references are known a priori, the system can react before the change has effectively been made inducing an anticipative effect, thus minimizing the effects of delay in the process response,
- Multivariable case can easily be dealt with,
- Extension to the treatment of the constraints (take into account actuators limitation),
- Development time relatively shorter than for competing advanced control methods.
- Easier to maintain and to implement in case of changing the model or its parameters. Indeed, model modification does not require complete redesign.

It also has its drawbacks:

- One of these is that the control law requires a longer computation time compared with conventional control strategies, *e.g.* Lyapunov-based design. When inequality constraints are considered, the amount of computation required is even higher,
- Generally, lack of stability proof except for special cases (as mentioned in the point above in the advantage section).

However, the greatest requirement is the need for an appropriate model of the process to be available. In fact, the design algorithm is based on a prior knowledge of the model but is not dependant on a specific model structure. It is however obvious that benefits obtained will be affected by the gap existing between the real process and the prediction model.

3.9 Concluding remarks

The receding horizon control is one of the most popular advanced control strategies due to its simplicity of implementation and interesting properties mentioned above. This chapter presented a general formulation of the model predictive control that will be used in the context of trajectory tracking problem. The brief review focused on the area of linear and nonlinear MPC schemes, addressing robustness issues and nominal and robust stabilities. The specific task we will focus on in the next chapter is the setpoint tracking NMPC for discrete-time nonlinear systems.

Chapter 4

Nonlinear Model Predictive Control

4.1 Introduction

In this chapter, we investigate the strategy that will be further considered and modified to solve a trajectory tracking problem for continuous/discrete-time nonlinear systems. This leads to explore the NMPC formulation discussed in Chapter 3. The chapter will be organized as follows. Section 4.2 is devoted to the NMPC problem formulation and the class of nonlinear systems that will be considered. A variant of NMPC is presented in section 4.3 and an illustrative example (a class of bioprocesses) is presented in section 4.4. Numerical results are provided in order to demonstrate the effectiveness and limitations of the NMPC strategy. Finally, conclusions are stated in section 4.5.

4.2 Problem formulation

This section will introduce the formulation of a basic standard nonlinear finite-horizon optimal control problem starting from a continuous-time model.

4.2.1 Continuous/discrete formulation

As mentionned in Chapter 3, the first step to implement the NMPC strategy is achieved with the use of a prediction model. In this context, it is important to obtain a mathematical representation as properly as possible reproducing the behavior of the system to be controlled. In our case, we will use a

continuous/discrete formulation: time-continuous model for prediction and discrete model for control.

Consider a system described by an uncertain continuous-time nonlinear model:

$$\begin{cases} \dot{x}(t) = F(x(t), u(t), \theta), \quad x_0 = \bar{x} \\ y(t) = Hx(t) \end{cases} \tag{4.1}$$

where

- $x \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ is the state vector with $\mathbb{X}$ the compact set of admissible states;
- $\bar{x}$ is the initial state vector;
- $y \in \mathbb{Y} \subseteq \mathbb{R}^{n_y}$ is the measured output with $\mathbb{Y}$ the compact set of admissible outputs;
- $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ represents the control input with $\mathbb{U}$ the compact set of admissible controls;
- $\theta \in \mathbb{R}^{n_\theta}$ is the vector of uncertain parameters that are assumed to lie in the compact set $\Theta = [\theta^-, \theta^+]$ defined as follows:

  $$\theta = \theta_{\text{nom}} + \delta\theta \tag{4.2}$$

  where θ_{nom} is the nominal parameters vector defined as the average value (centroid) of the compact set:

  $$\theta_{\text{nom}} = \frac{\theta^+ + \theta^-}{2} \tag{4.3}$$

  and $\delta\theta$ the parameters uncertainties vector;
- The mapping $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \longrightarrow \mathbb{R}^{n_x}$, of class $\mathcal{C}^1$ with respect to all its arguments, represents the nonlinear process dynamics;
- The measurement matrix is given by $H \in \mathbb{R}^{n_y \times n_x}$;

***Remark* 4.1.** The sets $\mathbb{X}$ and $\mathbb{U}$ are generally polyhedral convex sets, taking into account *e.g.* physical constraints acting on the system.

***Remark* 4.2.** The measurement could be nonlinear with respect to the state. It is assumed linear to simplifiy mathematical developments.

Exogenous inputs can act on system (4.1). They are omitted to simplify notation (but are applied to the system).
Most models of real life processes are given as continuous-time models, usually in the form of differential equations (4.1).

In order to convert these models for the design purposes, a continuous/discrete formulation is used.

The discrete-time outputs are obtained at each constant sampling time T_s by the integration of the continuous-time state space model (4.1) using for example the *Runge-Kutta* method with an integration time step T_d (lower than the sampling time T_s). The control input $u(t)$ is parametrized using a piecewise-constant approximation over a time interval $[t_k, t_{k+1}] \triangleq [kT_s, (k+1)T_s)]$:

$$u_k \triangleq u(\tau) = u(t_k), \quad \tau \in [t_k, t_{k+1}[\tag{4.4}$$

Let us define the discrete state trajectory g (see Figure 4.1) as the solution, at time t_{k+1}, of system (4.1):

$$\begin{cases} x_{k+1} = g(t_0, t_{k+1}, \bar{x}, u_{t_0}^{t_k}, \theta) \\ y_k = Hx_k \end{cases} \tag{4.5}$$

with initial state x_0, and $u_{t_0}^{t_k}$ the control sequence from the initial time instant t_0 to the time instant t_k. It can be obtained with a Runge-Kutta method for example.

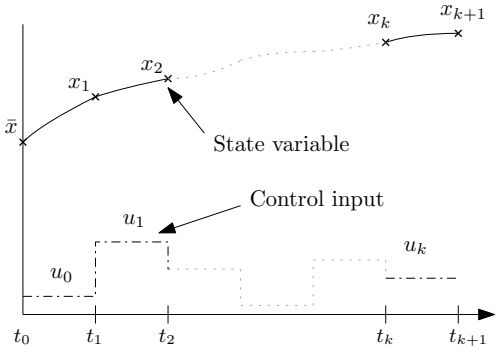

Figure 4.1: Discrete state path (SISO case).

Thus, the prediction model could be defined by the following recursive equations:

$$\begin{cases} x_{k+1} = f(x_k, u_k, \theta), k \geq 0, \quad x_0 = \bar{x} \\ y_k = Hx_k \end{cases} \tag{4.6}$$

where x_{k+1} is the state at time t_{k+1}, k is the time index, x_k and y_k are the discrete state vector and the sampled output at time t_k, respectively.

It can be easily shown that:

$$f(x_k, u_k, \theta) \equiv g(t_k, t_{k+1}, x_k, u_k, \theta) \tag{4.7}$$

In the sequel, the model (4.5) will be used in the NMPC strategy to predict the future behavior of the system.

4.2.2 Control objectives

In this study, the NMPC problem is formulated for trajectory tracking purposes. The main objective is to force the output signal y to follow a given reference trajectory y^r, while the control input u is constrained to track a reference u^r. In addition, saturations on the state vector and control input signal with minimum and maximum thresholds $x_{\min}$, $x_{\max}$, $u_{\min}$ and $u_{\max}$, respectively can be included (*i.e.* $\mathbb{X} = [x_{\min}, x_{\max}]$ and $\mathbb{U} = [u_{\min}, u_{\max}]$). These inequality constraints may result from both physical and operational constraints on the controlled system.

The predictive controller predicts the plant future evolution $\hat{y}_{k+1}^{k+N_p}$ over a finite time receding horizon of length $N_p T_s$, using a nonlinear dynamic model. At each time instant t_k, the optimal control sequence over the prediction horizon is computed by minimizing a cost function expressed as a quadratic criterion based on the future tracking errors, while ensuring that all constraints are respected. The first control in the optimal sequence is applied to the system until the next time step, when the measurement becomes available. The optimization problem is solved again at the next sampling time according to the well-known receding horizon principle [27, 90] as shown on Figure 4.2.

***Remark* 4.3.** This chapter is dedicated to the description of NMPC formalism, without consideration of robustness issues directly in the cost function. Therefore, even if uncertainties are included for simulation purposes, the system model will be considered as the nominal one with $\theta = \theta_{\text{nom}}$, for the derivation of the optimization problem.

4.2.3 Derivation of the control law

Following the general idea of the previous section, the cost function that will be minimized in this chapter is expressed as the sum of two quadratic functions based on tracking errors over the receding horizon and defined as (at time t_k):

$$\Pi_{\text{NMPC}}(x_k, u_k^{k+N_p-1}) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \tag{4.8}$$

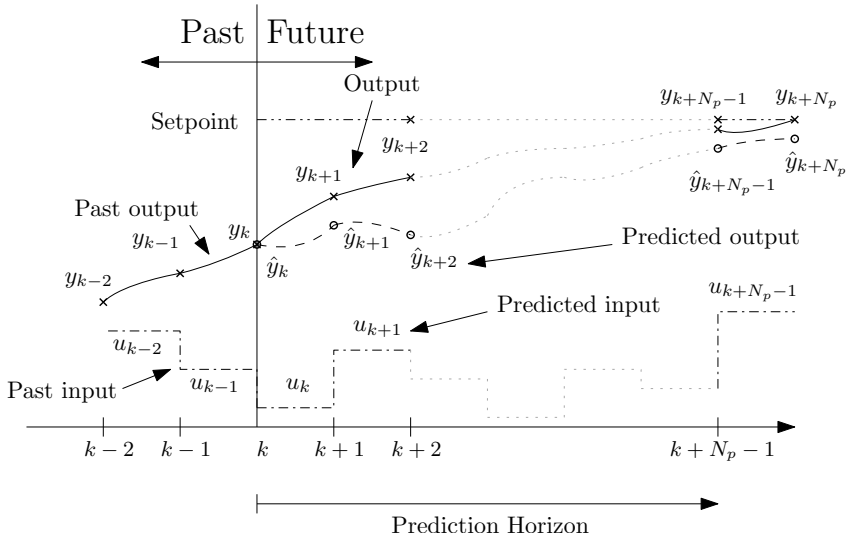

Figure 4.2: Principle of the NMPC (SISO case).

The discrete-time prediction model is chosen as:

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta) \\ \hat{y}_{k+j} = H\hat{x}_{k+j}, \quad \forall j = \overline{1, N_p} \end{cases} \tag{4.9}$$

subject to the constraints in a matrix form:

$$\begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \tag{4.10}$$

with x_k the state vector at time t_k,

$$u_k^{k+N_p-1} = \begin{bmatrix} u_k \\ \vdots \\ u_{k+N_p-1} \end{bmatrix} \text{ the optimization variable,}$$

$$u_k^{r,k+N_p-1} = \begin{bmatrix} u_k^r \\ \vdots \\ u_{k+N_p-1}^r \end{bmatrix} \text{ the reference control sequence,}$$

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, \theta_{\text{nom}}) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, \theta_{\text{nom}}) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, \theta_{\text{nom}}) \end{bmatrix} \text{ the predicted output sequence,} \tag{4.11}$$

and $y_{k+1}^{r,k+N_p} = \begin{bmatrix} y_{k+1}^r \\ \vdots \\ y_{k+N_p}^r \end{bmatrix}$ the setpoint values,
where the subscript is related to the time instant.

N_p is the length of the prediction horizon. $V \succeq 0$ and $W \succ 0$ are tuning weighting matrices.

As mentionned before, assuming a perfect knowledge of the parameter vector θ (*i.e.* $\theta = \theta_{\text{nom}}$, determined from an identification procedure for example), the formulation of the optimization problem is moved into NLP problem over a finite prediction horizon $N_p T_s$ at each sampling time t_k. The optimal control sequence is obtained by minimizing the performance criterion (4.8) with the constraints (4.10) as follows:

$$\overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \Pi_{\text{NMPC}}(x_k, u_k^{k+N_p-1}) \tag{4.12}$$

$$\text{s.t.} \quad (4.9)\text{-}(4.10)$$

This problem is in general non-convex, it is solved classicaly using algorithms for constrained optimization problems. However, in the case where the constraints are only bounds on the optimization variables, the optimization problem is more tractable.

***Remark* 4.4.** This formulation of the optimization problem belongs in fact to the sequential family of dynamic optimization methods (as mentioned in section 3.5).

***Remark* 4.5.** For the NMPC problem (4.12), at each sampling time $(k+1)T_s$ the optimization variable $u_{k+1}^{k+N_p}$ is initialized by the optimal control sequence $\overset{\star}{u}_{k+1}^{k+N_p-1}$ obtained from the optimization (4.12) at time instant k as follows:

$$\left(u_{k+1}^{k+N_p}\right)_{\text{ini}} = [\overset{\star}{u}_{k+1}^{k+N_p-1}, \ \overset{\star}{u}_{k+N_p-1}], \quad \forall k \geq 0$$

The NMPC problem is implemented as stated in Figure 4.3.

4.2.4 NMPC tuning

The computational burden of a NMPC algorithm is high and there are numbers of factors that affect it. These factors include the prediction horizon N_p, the optimization method, the number of states n_x and control variables

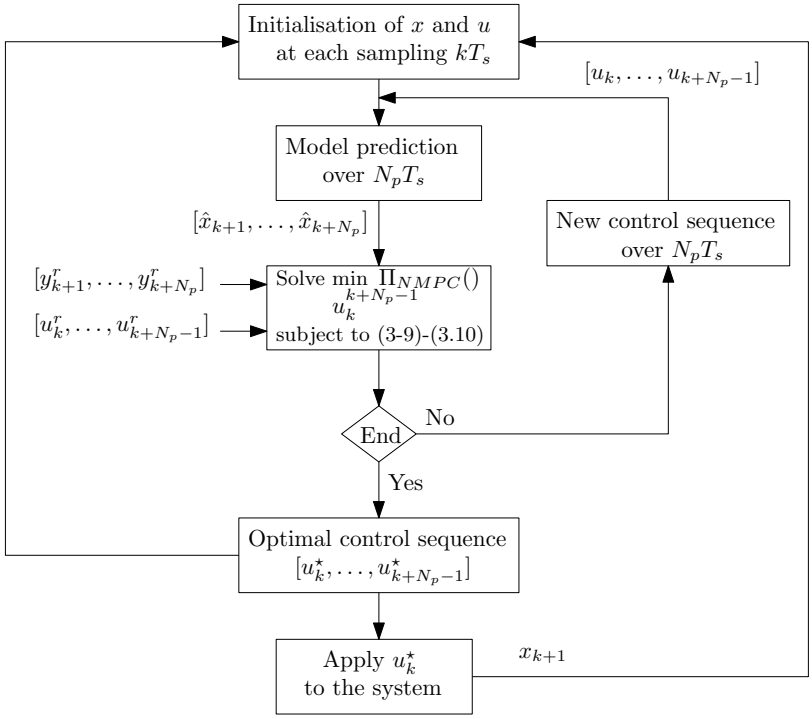

Figure 4.3: Diagram of the NMPC algorithm.

n_u, the sampling time step T_s, the weighting matrices V and W, the nonlinearity of the system (4.1), the calculation time of the model related to its complexity, and the speed of the computer processor. The main factors are discussed hereafter:

- **Prediction horizon:** the meaning of N_p is rather intuitive. It must be chosen in such a way that it gives a sufficient vision of the system behaviour. It marks the limit of the instants in which it is desirable for the output to follow the reference.

  ***Remark* 4.6.** Increasing the length of the prediction window N_p can lead to a loss of tracking accuracy. In fact, a large time window penalizes prediction when the model is far from the real one.

- **Weighting matrices:** the weights may be dictated by the economic objectives for the control system. Increasing the weighting matrix on the control action, V, relatively to the weighting matrix W has the effect of reducing the control effort and viceversa.

- **Sampling time:** in general, a major issue when solving an optimization problem is related to the discretization of the model (when using

a discrete/discrete formulation). Indeed, the validity of the discretized model requires choosing an adequate sampling time in agreement with the dynamics of the system. However, when solving a NLP problem, this sampling time must be large enough to permit online implementation. This is a trade-off to consider. In our case, we considered a continuous/discrete formulation. The choice helps reducing the impact of the sampling time value on the discrete model and thus on the controller stability. The choice of the sampling time will mainly condition the controller performance, leading to the same trade-off as mentionned previously (trade-off between computation time and controller performance). it should be mentionned that considering a small sampling time in general induces a large value of N_p, for a given prediction horizon.

- **Optimization method:** strategies for solving the NMPC problem are typically based on direct optimization methods using a finite parametrization of the input to find an open-loop solution to problem (4.12) that is implemented in a receding horizon framework. More details can be found in section 3.5.

4.3 A variant of NMPC

Model uncertainties can be taken into account by assuming that the gap between the system and the model at time instant k, denoted $\varepsilon^{s/m}$, is propagated over the prediction horizon, *i.e.* considering the accumulation of errors over j prediction intervals [141]. Thus, the predicted output of the system $\hat{y}_{k+j|k}$ is related to the predicted output through the model $\hat{y}^m_{k+j|k}$ after j prediction intervals through the following relation, where y_k denotes the measured output of the system at time instant kT_s:

$$\begin{cases} \hat{y}_{k+j|k} = \hat{y}^m_{k+j|k} + j\varepsilon^{s/m}_k, \quad j = \overline{1, N_p} \\ \varepsilon^{s/m}_k = y_k - \hat{y}^m_{k|k} \end{cases} \tag{4.13}$$

The term $j\varepsilon^{s/m}_k$ represents the integration of the modelling error up to time j. As a consequence, it could improve the control law performance with respect to model mismatch in comparison to including no error at all as presented before, *i.e.* $\hat{y}_{k+j|k} = \hat{y}^m_{k+j|k}$, or even including a constant error signal, *i.e.* we assume that the difference between the system and the model remains constant over the prediction horizon as in [127]:

$$\hat{y}_{k+j|k} = \hat{y}^m_{k+j|k} + \varepsilon^{s/m}_k \tag{4.14}$$

The structure of the NMPC law with consideration of the model-system error signal is given in Figure 4.4.

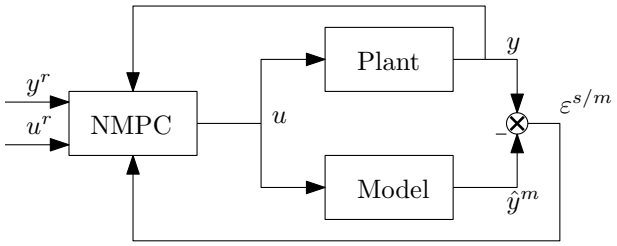

Figure 4.4: NMPC including $\varepsilon^{s/m}$ signal.

Although this strategy remains simple to implement and improves performances, even better results will be obtained in the following chapters by considering robust NMPC structures.

4.4 Numerical illustrative example

In the sequel, to illustrate the strategies presented previously, the control of a bacteria growth in a bioreactor is considered, in order to evaluate the performance of the setpoint tracking NMPC controller. Schematic representation of a bioreactor is shown in Figure 4.5. Generally, components are introduced into the bioreactor at the rate q_{in} and others are removed from the bioreactor at the rate q_{out}. Classically, three operating modes are used in practice:

- **Batch mode:** components are neither introduced nor removed *i.e.* $q_{in} = q_{out} = 0$.
- **Fed-batch mode:** only components are introduced ($q_{in} > 0$) and $q_{out} = 0$.
- **Continuous mode:** components are introduced and removed at the same rate *i.e.* $q_{in} = q_{out} > 0$.

Let us consider a biological process in continuous mode with a single biomass (bacteria) and a single substrate (nutrient). The dynamics of biomass and substrate concentrations, denoted X and S respectively, are obtained using mass balances for a continuous stirred tank reactor (CSTR) [9]. More details concerning the mass balance modelling can be found in Appendix B.

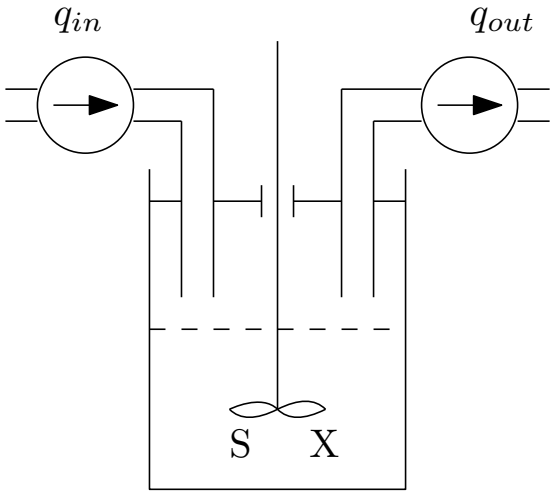

Figure 4.5: Schematic representation of a bioreactor.

The mass balances lead to the following system of ODE's that describes the microorganisms growth rate and the substrate consumption, respectively:

$$\begin{cases} \dot{X} = \underbrace{\mu(S)X}_{1} - \underbrace{DX}_{2} \\ \dot{S} = -\underbrace{\frac{\mu(S)}{Y}X}_{3} + \underbrace{DS_i}_{4} - \underbrace{DS}_{5} \end{cases} \tag{4.15}$$

with

1. The growth of biomass;
2. The dilution effect;
3. The consumption of substrate for the biomass growth;
4. The amount of incoming substrate into the reactor through the feed;
5. The dilution impact.

The dilution rate, D (h^{-1}), is defined as $D = \frac{q_{in}}{V} = \frac{q_{out}}{V}$. V is the volume of the bioreactor, S_i the inlet substrate concentration $(g.L^{-1})$, Y the yield production coefficient and $\mu(S)$ is the specific growth rate (h^{-1}).
The specific growth rate $\mu(S)$ is given by a Monod kinetic:

$$\mu(S) = \mu_m \frac{S}{S + K_m} \tag{4.16}$$

where μ_m and K_m are the maximal specific growth rate and semi-saturation constant, respectively.
The yield production coefficient, Y, represents the efficiency of conversion of substrate into biomass.
Following the state space formalism (4.1), the state vector x, the control input u and the parameter vector θ are given by:

$$x = \begin{bmatrix} X \\ S \end{bmatrix}, \quad u = D, \quad \theta = \begin{bmatrix} \mu_m \\ K_m \\ Y \end{bmatrix} \tag{4.17}$$

The control objective for the bioreactor given by (4.15) is to regulate the biomass concentration X (output variable) at a predetermined setpoint X^r, by manipulating the dilution rate D (control input) which remains closely to a specified reference, D^r_{nom}, while satisfying some accuracy requirements.

In order to illustrate the performance of the closed-loop system, numerical simulations were carried out by considering the values given in Table 4.1 for the parameters involved in the bioreactor model.

Table 4.1: Model parameters for system (4.15).

Parameter	Value	Unit
μ_m	0.3	h^{-1}
K_m	1.75	$g.L^{-1}$
Y	0.9	-

The simulation time is set to $T_f = 20$ hours. The reference trajectory X^r has two changes: a rising and falling edge respectively at two different instants as shown in Figure 4.6.
The reference dilution rate D^r_{nom} and the reference substrate concentration S^r_{nom} are determined at the equilibrium for the considered values of X^r and model parameters of Table 4.1, as follows (from (4.15) and (4.16)):

$$\begin{cases} S^r_{\text{nom}} = S_i - \dfrac{X^r}{Y} \\ D^r_{\text{nom}} = \mu_m \dfrac{S^r_{\text{nom}}}{S^r_{\text{nom}} + K_m} \end{cases} \tag{4.18}$$

In this nominal case, the biomass concentration X will follow its reference value X^r, while constraining the dilution rate D to track the reference D^r_{nom}.

It appears clearly that if the model is uncertain, the equilibrium for the considered values of X^r differs from the one given by (4.18). Indeed, the equilibrium corresponding to system under parameter uncertainties is determined

as follows:

$$\begin{cases} S^r_{\text{real}} = S_i - \dfrac{X^r}{Y + \delta Y} \\ D^r_{\text{real}} = (\mu_m + \delta\mu_m)\dfrac{S^r_{\text{real}}}{S^r_{\text{real}} + K_m + \delta K_m} \end{cases} \tag{4.19}$$

where $\delta\mu_m$, δK_m and δY are the parameters uncertainties.
In this case, the biomass concentration X should follow its reference value X^r, while constraining the dilution rate D to track the reference D^r_{real}. In section 4.4.2, the references S^r_{real} and D^r_{real} will be provided in order to better understand the behavior of the controller in the presence of parameters uncertainties.

***Remark* 4.7.** If there are no uncertainties, the equilibrium given by (4.19) is equivalent to the equilibrium given by (4.18).

The maximum admissible dilution rate $D_{\max}$ is set to $0.3\,h^{-1}$. The inlet substrate concentration S_i is assumed to be perfectly known ($35\ g.L^{-1}$). The initial conditions of the concentrations in the bioreactor are:

$$\begin{cases} S(0) = S^r_{\text{real}}(0)\ g.L^{-1} \\ X(0) = X^r(0)\ g.L^{-1} \end{cases} \tag{4.20}$$

The initial substrate concentration is set to the setpoint S^r_{real} in order to cancel the transient effect and to focus the study only on the behavior during the setpoint changes.
The time critical code including ODE's was written in C language using Matlab CMEX-functions. The optimization was run on Microsoft PC (Intel(R) Core(TM) $i7-3770$, 3.40 GHz, 8GB Ram). The optimization problem was solved using *lsqnonlin* from *Matlab* optimization toolbox.
Hereafter, the performance of the predictive controller is first studied in the nominal case (section 4.4.1) and then in the case of model parameters mismatch (section 4.4.2).

4.4.1 Influence of the tuning parameters

In this section, we shortly outline the influence of the prediction window N_pT_s and weighting matrices V and W on the closed-loop trajectory tracking behavior. For this purpose, we consider the NMPC as described above with the assumption that all states can be accessed directly by measurements. Simulations have been conducted considering the nominal case (*i.e.* without model mismatch).

First, the length of the prediction horizon and the weighting matrices are fixed at the values $N_p = 6$, $V = \mathbb{I}_{N_p}$ and $W = 0.01\mathbb{I}_{N_p}$ respectively. The

results are compared for three case studies concerning the choice of sampling time T_s (case 1: $T_s = 5$ min, case 2: $T_s = 10$ min and case 3: $T_s = 20$ min). The integration time step T_d is chosen $T_d = T_s/50$.

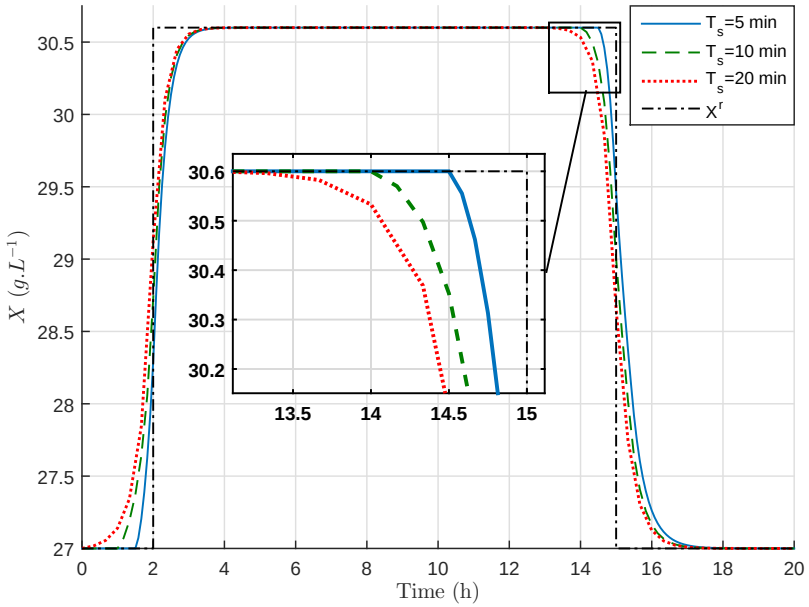

Figure 4.6: Influence of the sampling time T_s on the behavior of the biomass concentration for NMPC law.

It can be observed that the dilution rate D is decreased at time around 1h (Figure 4.8), this leads to increase the biomass concentration X in order to follow the desired setpoint X^r (Figure 4.6). Indeed, the dilution rate is decreased, so that it is lower than the biomass growth rate, leading to the increase of the micro-organisms concentration. When X reaches X^r, the controller sets D to D^r_{nom}, so that the biomass concentration is maintained at X^r. In this case, the substrate concentration decreases since it is consumed by the micro-organisms (Figure 4.7) and then it reaches its reference value. Furthermore, it can be noticed the anticipation behavior to a setpoint change, due to the prediction of the setpoint trajectory future evolution over the moving horizon. In the case of falling edge at time $t = 15\,h$, the dilution rate is set at its maximal value (Figure 4.8) to dilute the culture, X decreases as depicted by Figure 4.6 while S increases as shown in Figure 4.7. Then, when the setpoint is reached, the dilution rate is kept equal to its reference value, as well as the state variables X and S. The results as depicted in Figures 4.6-4.8 show that the larger sampling time leads to the larger prediction window. The model should be sampled sufficiently fast in order to guarantee its accuracy as much as possible but the sampling time T_s should be large

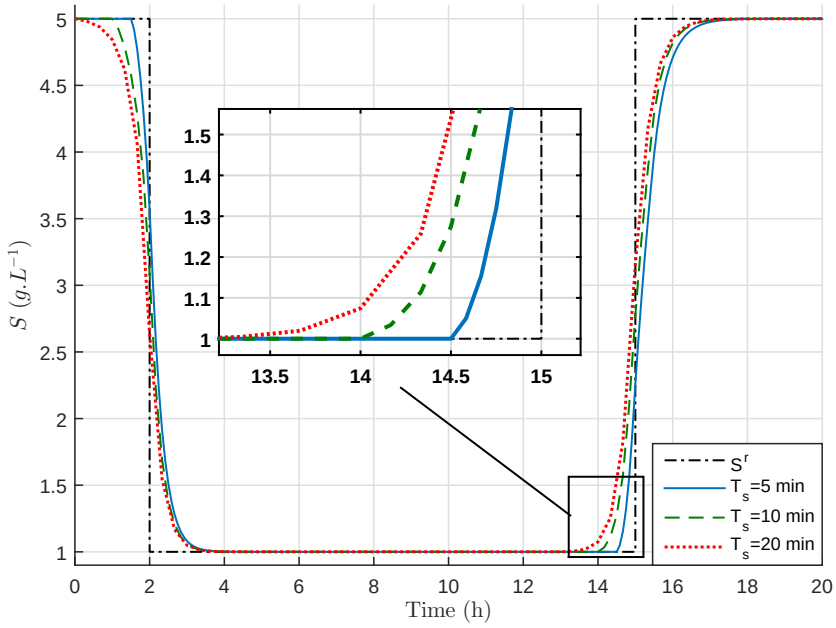

Figure 4.7: Influence of the sampling time T_s on the behavior of the substrate concentration for NMPC law.

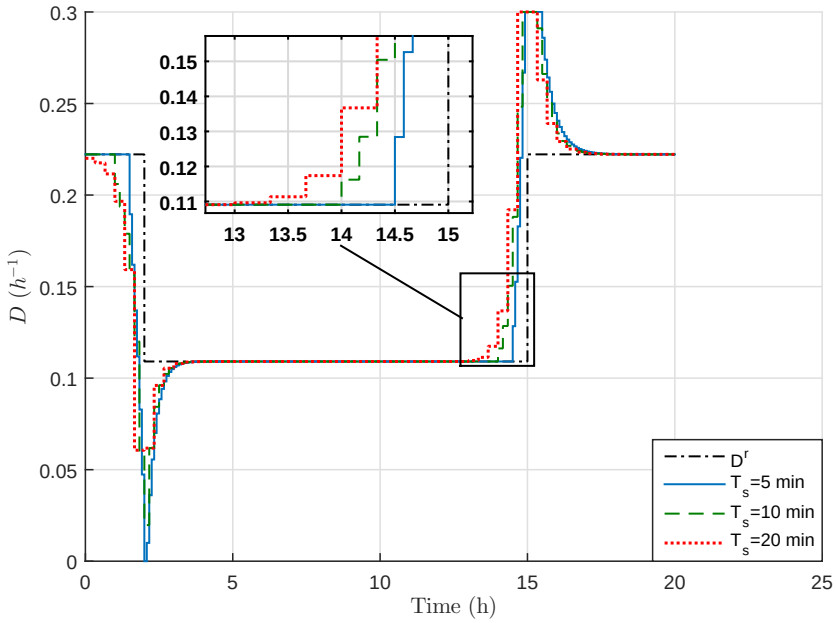

Figure 4.8: Influence of the sampling time T_s on the behavior of the control input for NMPC law.

enough to take into account the computation burden due to potential state estimator and/or online determination of the optimal trajectory. On the other hand, it can be noticed that the choice of T_s is linked to the size of the prediction window for a given value of the prediction horizon N_p. The sampling time T_s is hereafter chosen equal to 5 min, which is reasonable according to the dynamics of the system.

Secondly, the influence of the prediction horizon N_p is studied for the same conditions cited previously with $T_s = 5$ min, $V = \mathbb{I}_{N_p}$ and $W = 0.01\mathbb{I}_{N_p}$ as shown in Figures 4.9-4.11. The results are compared for three case studies (case 1: $N_p = 3$, case 2: $N_p = 6$ and case 3: $N_p = 12$). The integration time step will be given by $T_d = T_s/50$ and thus hereafter is set to 0.1 min.

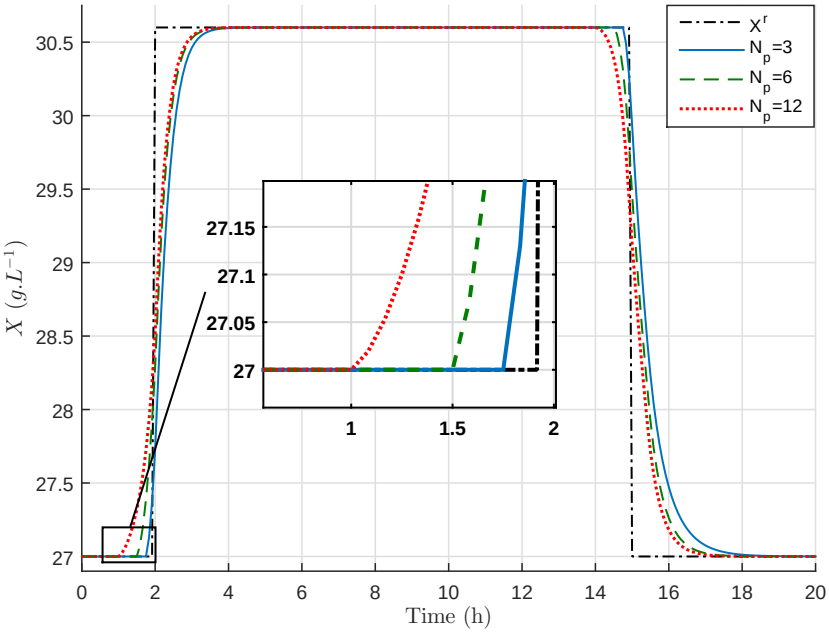

Figure 4.9: Influence of the prediction horizon N_p on the behavior of the biomass concentration for NMPC law.

It appears clearly that increasing the prediction horizon N_p leads to a larger anticipation time to a setpoint change as shown in Figure 4.9 (black box). The prediction horizon N_p must be chosen to satisfy a compromise between the computation time and a sufficient vision of the system behavior in the future. In fact, the dimension of the optimization problem increases with N_p leading to a greater computation time as shown in Table 4.2 (it provides computation times of iterations). Even if this time remains negligible compared to T_s in our case study, it can became critical for more complex systems with large number of parameters.

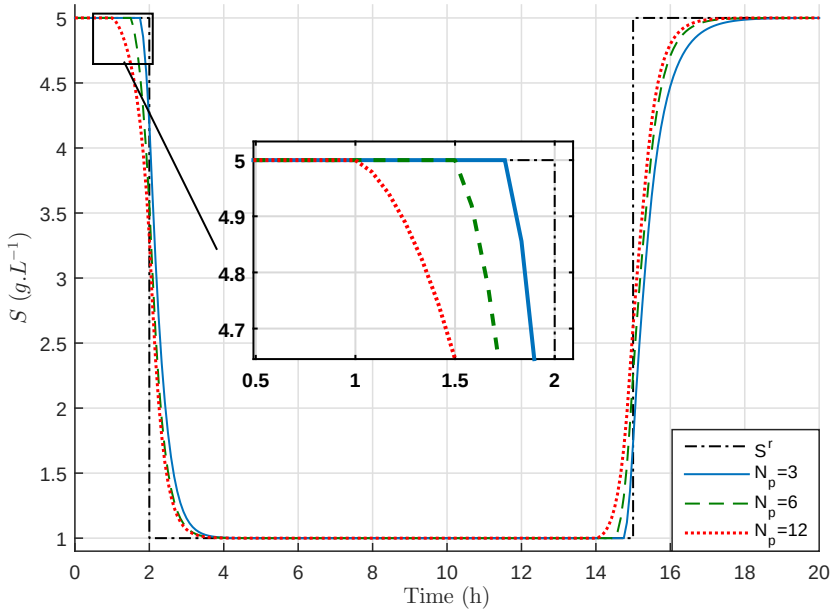

Figure 4.10: Influence of the prediction horizon N_p on the behavior of the substrate concentration for NMPC law.

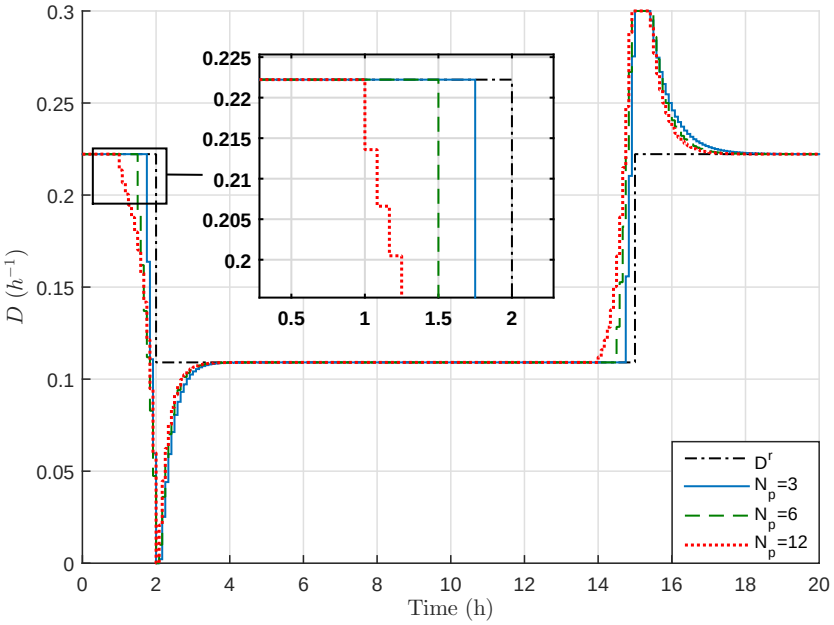

Figure 4.11: Influence of the prediction horizon N_p on the behavior of the control input for NMPC law.

Table 4.2: Influence of the prediction horizon N_p on the computation time.

	Computation time (s)		
Sampling time	min	mean	max
$N_p = 3$	$< 10^{-5}$	0.011	0.25
$N_p = 6$	$< 10^{-5}$	0.014	0.25
$N_p = 12$	$< 10^{-5}$	0.026	0.25

For this example, the prediction horizon is set to $N_p = 6$, which ensures a good trade-off between the computation time and a sufficient vision of the system behavior in the future.

Finally, the length of the prediction horizon, the sampling time and the integration time step are fixed at the values $N_p = 6$, T_s=5 min and T_d=0.1 min respectively. The results are compared for two case studies concerning the choice of weighting matrices (case 1: $V = W = \mathbb{I}_{N_p}$ and case 2: $V = \mathbb{I}_{N_p}$ $W = 0.01\mathbb{I}_{N_p}$). The matrix W is chosen less than one in order to emphasize the control tracking error in the cost function, and to induce a slight correction with the term related to the output tracking error (to smoothen the control input evolution). Thus, the choice of the weighting matrices directly conditions the state and control behaviors. They can be for example chosen based on the variables magnitude.

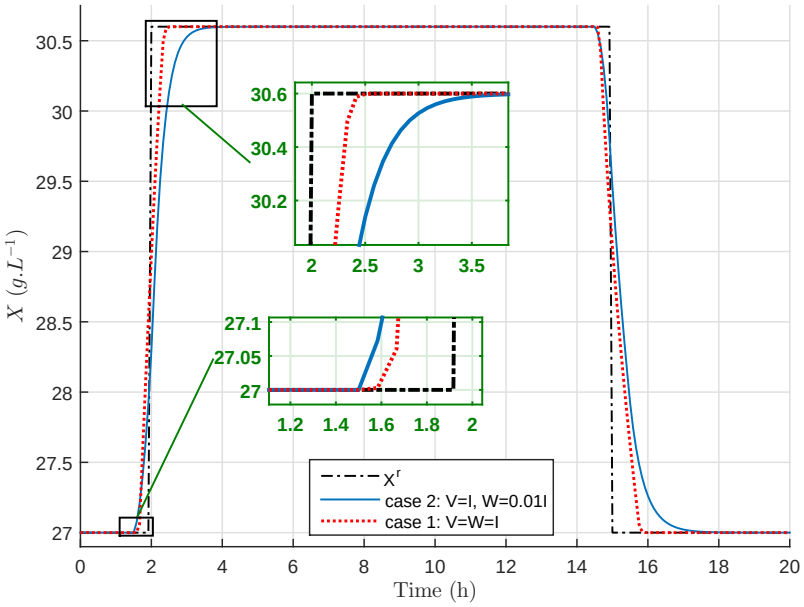

Figure 4.12: Influence of the weighting matrices on the biomass concentration evolution with time for NMPC law.

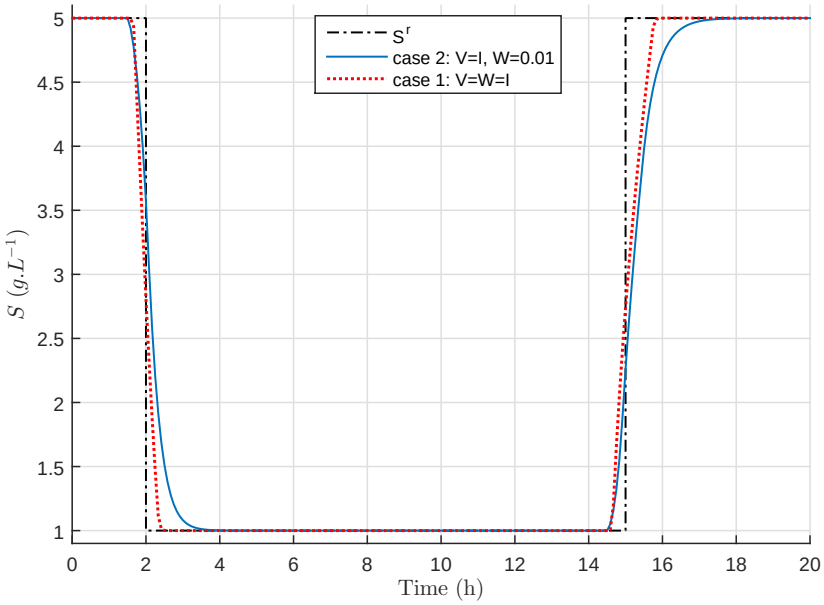

Figure 4.13: Influence of the weighting matrices on the substrate concentration evolution with time for NMPC law.

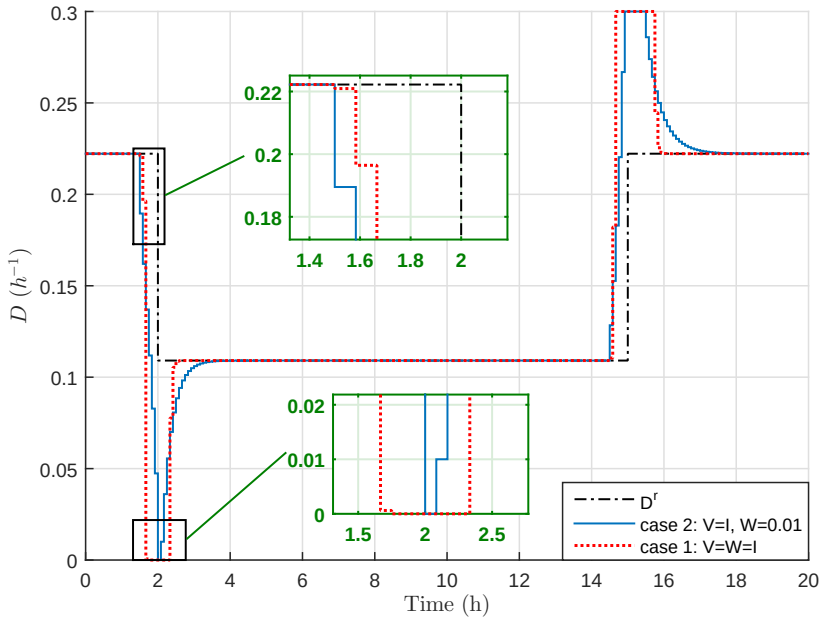

Figure 4.14: Influence of the weighting matrices on the control input evolution with time for NMPC law.

The obtained results are depicted in Figures 4.12-4.14. The chosen values

of the weights V and W in case 2 lead to an anticipation time of 30 min approximatively as shown in Figure 4.12 (green box) (this is also the case for case 1 even if their effect is here less visible). This result is in agreement with the size of the desired prediction window: N_pT_s=30 min. In the case 2, the control evolution is smoother and smaller, leading to a slightly longer response time. In order to ensure a quicker response time, the weighting matrices V and W will be chosen equal to $\mathbb{I}_{N_p}$ hereafter.

4.4.2 Influence of model parameters mismatch

In this section, the NMPC tuning parameters are chosen as follows: $N_p = 6$, T_s=5 min, T_d=0.1 min and $V = W = \mathbb{I}_{N_p}$.

Simulations were performed considering that the prediction model is erroneous. The maximum admissible dilution rate D_{max} is set to $0.5h^{-1}$ (it was set before at $0.3h^{-1}$ in order to emphasize the effects of the studied NMPC parameters and make them more visible). The parameters values of the system are chosen on the parameter subspace border referred to $\theta^{real} = [\mu_m^+ \; K_m^- \; Y^+]$. The parameters θ are assumed to be $\pm 20\%$ uncertain maximum. Figures 4.15-4.17 compare the results obtained when running simulation with and without addition of the error signal $j\varepsilon^{s/m}$ (see (4.13)) during prediction in the NMPC algorithm, as detailed in section 4.3. The addition of signal $\varepsilon^{s/m}$ (see (4.14)) is also considered. The controller that includes the $j\varepsilon^{s/m}$ signal is denoted NMPC-$j\varepsilon^{s/m}$ whereas the one that includes $\varepsilon^{s/m}$ is denoted NMPC-$\varepsilon^{s/m}$.
The dilution rate moves away from its reference value D^r_{nom} (calculated with the nominal model (4.18)) and converges to D^r_{real} (calculated with (4.19)) as shown in Figure 4.17.
It can be seen that NMPC-$j\varepsilon^{s/m}$ has better performances than the classical NMPC under parameters uncertainties. It can be noticed that without the $(j)\varepsilon^{s/m}$ term, the output is not able to track the specified setpoint in the presence of parameters uncertainties, due to the fact that the gap between the system and the model is not considered during the prediction step inside the minimization procedure. Taking the signal $j\varepsilon^{s/m}$ into account during the optimization strategy helps to improve significantly the tracking accuracy in comparison with when considering a constant signal $\varepsilon^{s/m}$ (Figure 4.15). This is due to the fact that the control input of NMPC-$j\varepsilon^{s/m}$ converges exactly to D^r_{real} which is not the case for NMPC-$\varepsilon^{s/m}$.

Thus, the predictive controller regulates the biomass to its reference value with good performances in the nominal case. It however lacks accuracy in the case of model mismatch. This example will be considered in the next chapters to evaluate the performance of the proposed predictive algorithms.

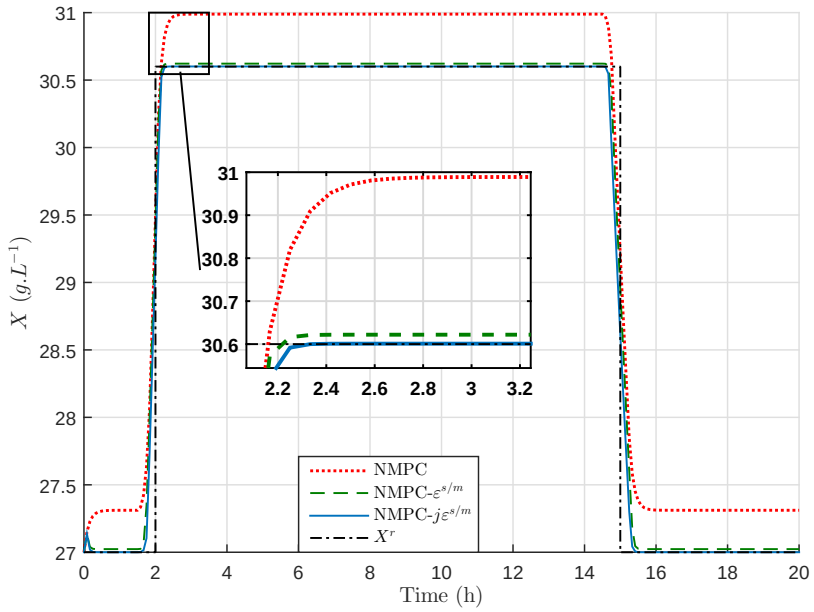

Figure 4.15: Biomass concentration evolution with time for NMPC-$(j)\varepsilon^{s/m}$ laws in the case of parameter uncertainties.

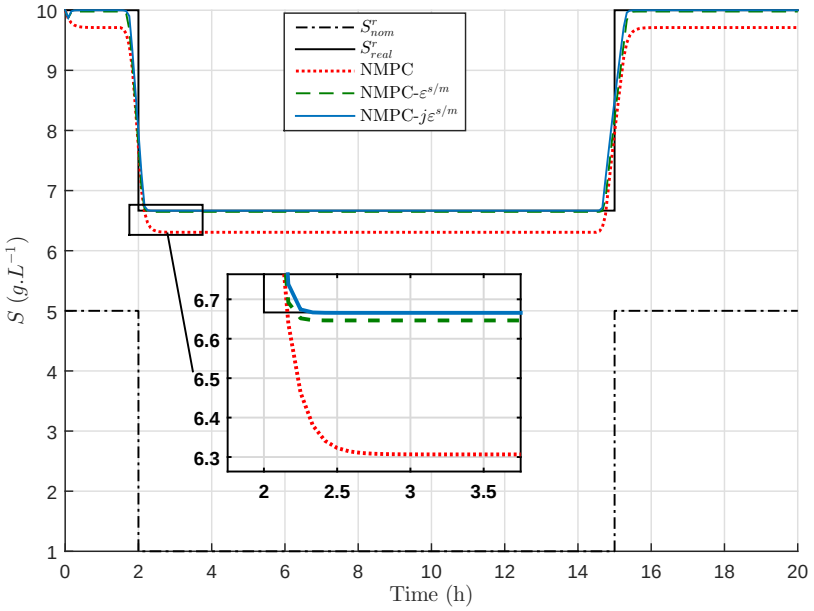

Figure 4.16: Substrate concentration evolution with time for NMPC-$(j)\varepsilon^{s/m}$ laws in the case of parameter uncertainties.

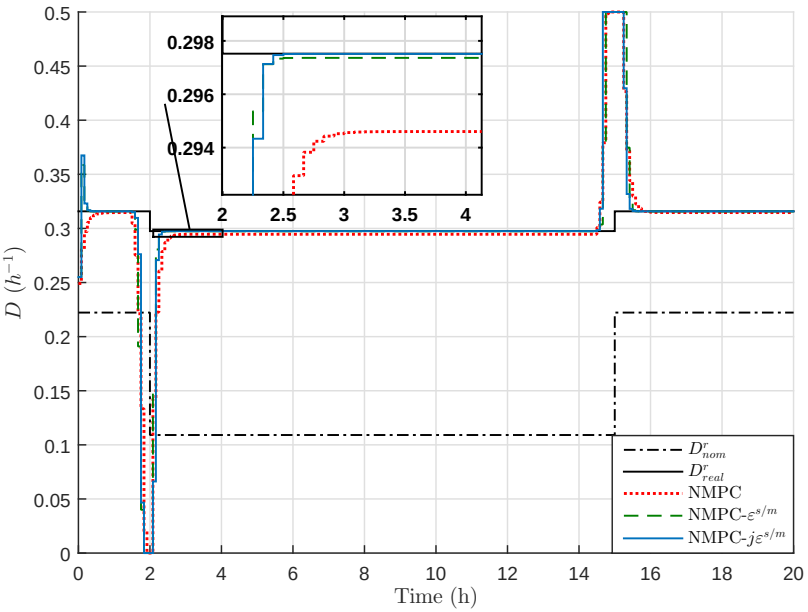

Figure 4.17: Control input evolution with time for NMPC-$(j)\varepsilon^{s/m}$ laws in the case of parameter uncertainties.

4.5 Concluding remarks

In this chapter, we presented the NMPC problem formulation for setpoint tracking purpose. This strategy induces solving, online, an optimization problem which is expressed as a nonlinear programming problem. The output signal is not able to track accurately the specified setpoint in the presence of parameters uncertainties, due to the fact that the mismatch between the system and the model is not considered during the prediction step inside the minimization procedure. Consequently, there is a need for improvement in the case of parameters uncertainties. One improvement could be to include, during the prediction step, the difference between the system and the model outputs in order to increase the control law robustness against model mismatch. Several simulations were performed in order to compare the predictive strategy performances for an illustrative biological application. They show the efficiency of the proposed control law, but also highlighted its limits. The assumption on including the prediction error is quite restrictive due to the fact that we do not exactly know, in a realistic application, how the error will evolve over the prediction horizon. If the assumption is false then the performances will be degraded. The drawbacks of this method are a motivation to investigate a new approach in order to satisfy the objectives of robustness. An optimization problem taking into account the model pa-

rameters uncertainties must be formulated in order to ensure the robustness against model mismatch. The next chapter will be then devoted to the robust model predictive control.

Chapter 5

Robust Nonlinear Model Predictive Control

5.1 Introduction

In the previous chapter, a predictive controller was designed to ensure a stable real time operation of the plant, close to a certain state or desired profile. However, the performances of the NMPC law usually decrease when the true plant evolution deviates significantly from the one predicted by the model. Robust variants of NMPC [78, 86] as detailed in section 3.6 are able to take into account set bounded disturbance and/or constraints. In addition, when considering uncertain systems, or when only a limited amount of data is available, it is necessary to use a robust formulation of the controller where the effect of the uncertainties can be taken into account in the design procedure. The robust NMPC (RNMPC) law can be formulated as a nonlinear min-max optimization problem (where the objective function is minimized for the worst possible uncertainty realization). However, this approach tends to become too complex to be solved numerically online. Consequently, the total calculation time is an important factor that must be reduced as much as possible. We propose mainly two solutions to address this problem. The first one consists in reducing the number of uncertain parameters through a sensitivity analysis. Only the main influencial parameters on the model are considered in the min-max problem whereas the other parameters are kept constant and equal to their nominal values. This approach will be referred to as reduced RNMPC (rRNMPC). The second approach aims at transforming the min-max problem into a more tractable optimization problem using a model linearization technique. The objective is then to propose a new formulation that is computationally more tractable in calculating the opti-

mal control compared to a classical min-max robust approach, which makes it suitable for online implementation. This approach will be referred to as Linearized RMPC (LRMPC).

Stability properties of the robust model predictive control strategy taking into account bounded uncertainties has been analyzed in [94, 99, 120, 119, 100, 19, 137]. Based on work developped by [86] and [120], and taking the objective function as the Lyapunov candidate function, the robust stability of the closed-loop system while applying the LRMPC law is established under some assumptions.

The LRMPC controller represents our main contribution. Thus, the stability and performances analysis will be focused on this approach.

The chapter is organized as follows. The next section is devoted to the min-max problem RNMPC formulation. A variant of RNMPC, based on a reduction of the number of the parameters θ that will be optimized, is presented in section 5.3. Section 5.4 is dedicated to the use of a model linearization technique in order to approach the min-max problem. Stability analysis of the closed-loop system is also discussed. The unconstrained formulation (LRMPC) is detailed in section 5.5. In section 5.6, a robust predictive controller (CLRMPC), similar to the above one with consideration of inequality constraints on the control input, is presented. Numerical results are provided in section 5.7 and conclusions are stated in section 5.8.

5.2 Min-max strategy

Since the predictive controller is model-based, it is very sensitive to model uncertainties, and more specifically to the model parameters values. In this context, we will assume that the parameter vector θ is uncertain and belongs to a known set Θ as stated in section 4.2.1. Thus, robustification in the presence of model uncertainties naturally leads to the formulation of a nonlinear min-max optimization problem [4, 93, 102]. The control sequence that minimizes a worst case cost function is derived from the following optimization problem (at time index k and using the same notation as in section 4.2):

$$\overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\theta} \Pi_{\text{RNMPC}}(x_k, u_k^{k+N_p-1}, \delta\theta) \tag{5.1}$$

subject to

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta = \theta_{\text{nom}} + \delta\theta), \;\; j = \overline{1, N_p} \\ \theta \in \Theta \\ \begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \end{cases} \tag{5.2}$$

where θ and θ_{nom} are given in (4.2) and (4.3).
The cost function associated to the future evolution of the system depends on the future control actions and the uncertainties as follows:

$$\Pi_{\text{RNMPC}}(x_k, u_k^{k+N_p-1}, \delta\theta) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \tag{5.3}$$

The predicted outputs $\hat{y}_{k+1}^{k+N_p}$ having the same formulation as in (4.11):

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, \theta) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, \theta) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, \theta) \end{bmatrix} \tag{5.4}$$

Relations (5.2) and (5.3) are thus similar to the classical NMPC approach, but by considering a value for θ given by the optimization algorithm instead of θ_{nom}.
The optimal control sequence $\overset{\star}{u}_k^{k+N_p-1}$ is determined to minimize the tracking error by considering all trajectories over all possible data scenarii [54, 76].
The RNMPC problem is implemented as depicted in Figure 5.1.

***Remark* 5.1.** For the min-max problem (5.1), at sampling time $(k+1)T_s$ the optimization variables $u_{k+1}^{k+N_p}$ and $\delta\theta$ are initialized by the optimal control sequence $\overset{\star}{u}_{k+1}^{k+N_p-1}$ and the optimal parameter vector $\overset{\star}{\delta\theta}$ obtained from the optimization (5.1) at time instant k as follows:

$$\left(u_{k+1}^{k+N_p}\right)_{\text{ini}} = [\overset{\star}{u}_{k+1}^{k+N_p-1}, \;\; \overset{\star}{u}_{k+N_p-1}], \quad \forall k \geq 0$$

$$\delta\theta_{\text{ini}} = \overset{\star}{\delta\theta} = \arg \max_{\delta\theta} \; \Pi_{\text{RNMPC}}(x_k, \overset{\star}{u}_k^{k+N_p-1}, \delta\theta)$$

***Remark* 5.2.** It appears clearly that the computation load grows with the size of the parameters vector, the number of control inputs and the prediction horizon, while the control strategy has to be implemented online. The challenge is to reduce the computation burden while maintaining good performances in term of accuracy.

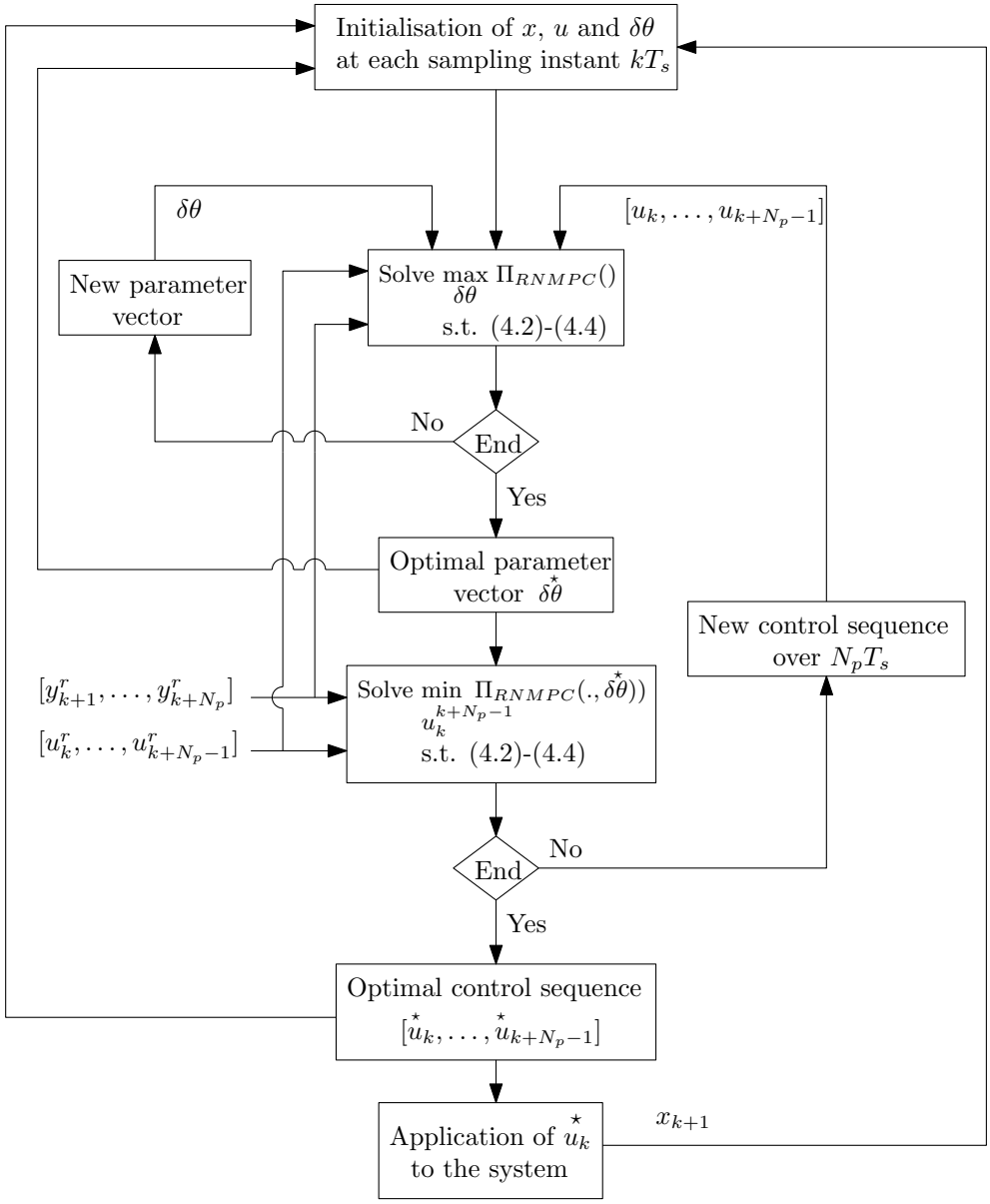

Figure 5.1: Diagram of RNMPC based on min-max optimization problem.

5.3 Reduced robust predictive controller

Since the initial min-max optimization problem (5.1)-(5.3) is time consuming, it will be simplified by reducing the number of the parameters θ that will be optimized from a sensitivity analysis of the model with respect to its parameters.

5.3.1 Sensitivity analysis

The sensitivity functions $S_{\theta_j}^{x_i}$ represent the sensitivity of each state x_i to (small) variations in each model parameter θ_j. It is defined as the partial derivatives of the state variable vector x_i with respect to θ_j:

$$S_{\theta_j}^{x_i} \triangleq \frac{\partial x_i}{\partial \theta_j}, \quad i = \overline{1, n_x} \text{ and } j = \overline{1, n_\theta} \tag{5.5}$$

Different approaches are possible to determine the sensitivity functions. The most precise method involves analytical derivation [52]. In this case, the dynamics of sensitivities are calculated as follows for system (4.1):

$$\dot{S}_{\theta_j}^{x_i} = \frac{d}{dt}\left(\frac{\partial x_i}{\partial \theta_j}\right) = \frac{\partial}{\partial \theta_j}\left(\frac{dx_i}{dt}\right) = \frac{\partial F_i}{\partial \theta_j} + \sum_{k=1}^{n_x} \frac{\partial F_i}{\partial x_k}\left(\frac{\partial x_k}{\partial \theta_j}\right) \tag{5.6}$$

with as an initial condition: $\frac{\partial x_i}{\partial \theta_j} = 0$.
From the analysis of the sensitivity functions temporal evolution, and according to their magnitude order, one can select the parameters which are significantly the most influential on the model [52]. These parameters will be denoted κ.

5.3.2 Problem reformulation

In the sequel, only the most influential parameters, $\kappa \in \Theta_\kappa \subseteq \Theta$, are considered in the min-max optimization, instead of the full model parameters (with $\theta \triangleq [\kappa, \zeta]$). The influential parameters κ are defined as follows:

$$\begin{cases} \kappa = \kappa_{\text{nom}} + \delta\kappa & (5.7) \\ \kappa_{\text{nom}} = \dfrac{\kappa^+ + \kappa^-}{2} & (5.8) \end{cases}$$

where κ_{nom} is the nominal influential parameters and $\delta\kappa$ the parameters uncertainties vector. The other parameters, ζ, are set to their nominal values with $\zeta_{nom} = (\zeta^+ + \zeta^-)/2$.

Thanks to the sensitivity analysis, the min-max optimization (5.1)-(5.3) is approached by the following optimization problem (at time index k):

$$\overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\kappa} \Pi_{\text{rRNMPC}}(x_k, u_k^{k+N_p-1}, \delta\kappa) \tag{5.9}$$

subject to

$$\begin{cases} \hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \begin{bmatrix} \kappa_{\text{nom}} + \delta\kappa \\ \zeta_{nom} \end{bmatrix}), \;\; j = \overline{1, N_p} \\ \kappa \in \Theta_\kappa \\ \begin{bmatrix} \mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ -\mathbb{I}_{n_x} & \mathbb{0}_{n_u} \\ \mathbb{0}_{n_x} & \mathbb{I}_{n_u} \\ \mathbb{0}_{n_x} & -\mathbb{I}_{n_u} \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j} \\ u_{k+j-1} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix}, \quad \forall j = \overline{1, N_p} \end{cases} \tag{5.10}$$

where κ_{nom} and $\delta\kappa$ are given above.
The new cost function is defined as follows:

$$\Pi_{\text{rRNMPC}}(u_k^{k+N_p-1}, \delta\kappa) \triangleq ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + ||\hat{y}_{k+1}^{k+N_p} - y_{k+1}^{r,k+N_p}||_W^2 \tag{5.11}$$

The predicted outputs $\hat{y}_{k+1}^{k+N_p}$ are given by:

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \\ Hg(t_k, t_{k+2}, x_k, u_k^{k+1}, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, [\kappa^\top, \;\; \zeta_{nom}^\top]^\top) \end{bmatrix} \tag{5.12}$$

This approach presents however some drawbacks:

- A sensitivity analysis of the model with respect to it parameters is required, leading to a more complex and time demanding developments.
- For some systems, all parameters can have high influence on the model, and thus no parameter reduction is possible (*e.g.* for systems with a small number of parameters).

Consequently, this approach cannot be used in all cases (it is then equivalent to the RNMPC). Hereafter, this approach is referred to as reduced RNMPC (rRNMPC).

5.4 Linearized robust predictive controller

Since the min-max optimization problem (5.1)-(5.4) even when reducing the number of uncertain parameters as in (5.9)-(5.12), is time consuming, it will be approached further by a more tractable optimization problem. This step focuses on reducing the computational burden of the initial problem. The

key idea is to approach the predicted outputs based on the nonlinear model through linearization technique.

As a direct result of the linearization, the non-convex problem will be approached by a convex one. So, more flexible tools will be accessible to better handle the optimization problem.

5.4.1 Main principle

In the following proposed approach, the outputs in the moving time frame are predicted by Taylor series expansion. A similar dual problem for robust state estimation, consisting in the design of receding-horizon observer, was presented in [61]. In this section, we propose to apply the same approach to the case of NMPC law design.

Based on a first order Taylor series expansion of (4.7), the prediction model g for time t_{k+1}, starting from state x_k, is linearized around the reference trajectory given by the reference control u_k^r and for the nominal parameters θ_{nom}:

$$\hat{x}_{k+1} \approx \hat{x}_{\text{nom},k+1} + \nabla_\theta g(t_{k+1})\delta\theta + \nabla_u g(t_{k+1})(u_k - u_k^r) \triangleq f_p(x_k, u_k, \theta_{\text{nom}} + \delta\theta) \tag{5.13}$$

with

$$\begin{cases} \delta\theta = \theta - \theta_{\text{nom}} & (5.14) \\ \hat{x}_{\text{nom},k+1} = g(t_k, t_{k+1}, x_k, u_k^r, \theta_{\text{nom}}) & (5.15) \\ \nabla_\theta g(t_{k+1}) = \left.\dfrac{\partial g(t_k, t_{k+1}, x_k, u_k, \theta)}{\partial \theta}\right|_{\substack{u_k = u_k^r \\ \theta = \theta_{\text{nom}}}} & (5.16) \\ \nabla_u g(t_{k+1}) = \left.\dfrac{\partial g(t_k, t_{k+1}, x_k, u_k, \theta)}{\partial u_k}\right|_{\substack{u_k = u_k^r \\ \theta = \theta_{\text{nom}}}} & (5.17) \end{cases}$$

Generalizing, the predicted state for time t_{k+j}, starting from state at t_k, is linearized around the reference trajectory given by the reference control sequence $u_k^{r,k+j-1}$ and for θ_{nom}. Using the same approach as in (5.13) for $j = \overline{1, N_p}$, it comes:

$$\hat{x}_{k+j} \approx \hat{x}_{\text{nom},k+j} + \nabla_\theta g(t_{k+j})\delta\theta + \nabla_u g(t_{k+j}) \left(u_k^{k+j-1} - u_k^{r,k+j-1}\right) \tag{5.18}$$

with

$$\begin{cases} \hat{x}_{\text{nom},k+j} = g(t_k, t_{k+j}, x_k, u_k^{r,k+j-1}, \theta_{\text{nom}}) & (5.19) \\ \nabla_\theta g(t_{k+j}) = \left.\dfrac{\partial g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta)}{\partial \theta}\right|_{\substack{u_k^{k+j-1} = u_k^{r,k+j-1} \\ \theta = \theta_{\text{nom}}}} & (5.20) \\ \nabla_u g(t_{k+j}) = \left.\dfrac{\partial g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta)}{\partial u_k^{k+j-1}}\right|_{\substack{u_k^{k+j-1} = u_k^{r,k+j-1} \\ \theta = \theta_{\text{nom}}}} & (5.21) \end{cases}$$

Different approaches are possible for determining the sensitivity functions with respect to the parameters vector and the control sequence, defined in (5.20) and (5.21), respectively as presented in section 5.3.1.

The dynamics of the sensitivity function with respect to θ can be computed for time $t \in [t_k, t_{k+N_p}]$ by solving numerically the following differential equation (from (5.6)):

$$\frac{d}{dt}\left(\nabla_\theta g(t)\right) = \frac{\partial F(x(t), u(t), \theta_{\text{nom}})}{\partial x} \nabla_\theta g(t) + \frac{\partial F(x(t), u(t), \theta)}{\partial \theta}|_{\theta=\theta_{\text{nom}}} \quad (5.22)$$

with as an initial condition:

$$\nabla_\theta g(t_k) = \mathbb{0}_{n_x \times n_\theta} \quad (5.23)$$

Indeed, $\nabla_\theta g$ is the sensitivity function of the state x with respect to the parameter θ.

An alternative procedure is to use the finite differences in order to approximate numerically the derivatives $\nabla_\theta g$ for each parameter θ_l, $l \in [1, n_\theta]$ and $\nabla_u g(t_{k+j})$ for each control u_j, $j \in [k, k + N_p - 1]$.

The finite difference method approximates the (i, j)-th element of the jacobian of a vector function $g(z)$ as

$$\nabla g(z) \approx \frac{g_i(z_j + \delta) - g_i(z_j)}{\delta} \quad (5.24)$$

for some small $\delta > 0$. A too large δ will induce inaccuracies due to the nonlinearity of g_i, since the method computes the average slope between two points.

***Remark* 5.3.** The most accurate result and computationally most efficient approach is to calculate gradients analytically (by symbolic differentiation). Doing this by hand, or even using symbolic computations in *Maple* or *Mathematica*, may quickly become intractable for MPC problems that may contain a large number of variables and parameters.

In order to simplify the calculation of the gradients $\nabla_\theta g$ and $\nabla_u g$, finite differences are used to approximate numerically the derivatives $\nabla_\theta g(t_{k+j})$ and $\nabla_u g(t_{k+i})$.
From (4.11) and (5.18), the predicted ouputs over the moving horizon are expressed as follows:

$$\hat{y}_{k+1}^{k+N_p} = G_{\text{nom},k+1}^{k+N_p} + G_{\theta,k+1}^{k+N_p}\delta\theta + G_{u,k}^{k+N_p-1}(u_k^{k+N_p-1} - u_k^{r,k+N_p-1}) \tag{5.25}$$

where

$G_{\text{nom},k+1}^{k+N_p} = \begin{bmatrix} H\hat{x}_{\text{nom},k+1} \\ \vdots \\ H\hat{x}_{\text{nom},k+j} \\ \vdots \\ H\hat{x}_{\text{nom},k+N_p} \end{bmatrix}$, is the column vector containing the predicted output for the nominal case.

$G_{\theta,k+1}^{k+N_p} = \begin{bmatrix} H\nabla_\theta g(t_{k+1}) \\ \vdots \\ H\nabla_\theta g(t_{k+j}) \\ \vdots \\ H\nabla_\theta g(t_{k+N_p}) \end{bmatrix}$, regroups the Jacobian matrices related to the parameters.

$G_{u,k}^{k+N_p-1} = \begin{bmatrix} H\nabla_u g(t_{k+1}) \\ \vdots \\ H\nabla_u g(t_{k+j}) \\ \vdots \\ H\nabla_u g(t_{k+Np}) \end{bmatrix}$, regroups the Jacobian matrices related to the control sequence.
Assuming that the uncertain parameters are uncorrelated and recalling that

$$\theta^- \leq \theta \leq \theta^+ \tag{5.26}$$

and

$$\theta_{\text{nom}} = \frac{\theta^+ + \theta^-}{2} \tag{5.27}$$

Thus,

$$\frac{\theta^- - \theta^+}{2} \leq \theta - \theta_{\text{nom}} \leq \frac{\theta^+ - \theta^-}{2} \tag{5.28}$$

Then, the bounded parametric error $\delta\theta$ can be expressed by:

$$\delta\theta = \gamma\delta\theta_{\max} \tag{5.29}$$

with

$$\delta\theta_{\max} = (\theta^+ - \theta^-)/2 \tag{5.30}$$

and

$$||\gamma|| \leq 1 \tag{5.31}$$

The initial objective function Π_{RNMPC} (5.3) is substituted by a cost function using the equation (5.25). The result is given by the following expression (with the same notations as in (4.11)):

$$\begin{aligned} &\Pi_{\text{RNMPC}}(x_k, u_k^{k+N_p-1}, \delta\theta) \approx ||u_k^{k+N_p-1} - u_k^{r,k+N_p-1}||_V^2 + \\ &||G_{\text{nom},k+1}^{k+N_p} - y_{k+1}^{r,k+N_p} + G_{\theta,k+1}^{k+N_p}\delta\theta + G_{u,k}^{k+N_p-1}(u_k^{k+N_p-1} - u_k^{r,k+N_p-1})||_W^2 \\ &\qquad \triangleq \Pi(x_k, u_k^{k+N_p-1}, \delta\theta) \end{aligned} \tag{5.32}$$

The new optimization problem is given by:

$$\overset{\star}{u}_k^{k+N_p-1} = \arg \min_{u_k^{k+N_p-1}} \max_{\delta\theta} \Pi(x_k, u_k^{k+N_p-1}, \delta\theta) \tag{5.33}$$

subject to

$$\begin{cases} \theta \in \Theta, \quad x \in \mathbb{X}, \quad u \in \mathbb{U} \\ \delta\theta = \gamma\delta\theta_{\max} \end{cases} \tag{5.34}$$

5.4.2 Stability analysis

In this section, the robust stability of the closed-loop system (4.6) with (5.33)-(5.34) is analysed by exploiting the results obtained in [19, 120, 86].

More specifically, we will use the following theorem:

***Theorem* 5.1.** Consider a discrete-time nonlinear system given by:

$$x_{k+1} = l(x_k, w_k), \quad k \geq 0, \quad x_0 = \bar{x} \tag{5.35}$$

where $x_k \in \mathcal{X}$ is the state of the system, $w_k \in \mathcal{W}$ is the disturbance vector (s.t. $\mathcal{W}$ is a compact set that contains the origin).
If system (5.35) admits a robust Lyapunov function, then it is robustly stable.

Proof. see [86].

We adapt the results in the above cited references to address the stability analysis of the proposed control strategy. First, a bound on the prediction error will be determined. Secondly, we give an upper and lower bounds on

the optimal cost. Finally, we establish the robust stability of the closed-loop system.

In the sequel, for the stability analysis of the closed-loop system (4.6) with (5.1)-(5.32), we need to consider the following assumptions:

***Assumption* 5.1.** The state of the plant x_k is measured at each sampling time.

***Assumption* 5.2.** The state x and the control u of the plant must fulfill the following constraints: $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$. where $\mathbb{X}$ and $\mathbb{U}$ are compact sets, both of them contain the origin.

5.4.2.1 Bound on prediction error

In this subsection, an upper bound on the prediction error provided by the linearization step is derived. Consider the real system for time t_{k+1}, starting from state x_k at time t_k:

$$x_{k+1} = f(x_k, u_k, \theta_{\text{nom}} + \delta\theta^s) \tag{5.36}$$

where θ_{nom} is the nominal parameters vector and $\delta\theta^s$ the real parameter values mismatch.

***Assumption* 5.3.** The function f is of class C^2 with respect to all its arguments.

From *Assumption* 5.3, and using Taylor developments (around θ_{nom} and u_k^r), the system dynamics can be rewritten as follows:

$$\begin{aligned} x_{k+1} = {} & f(x_k, u_k^r, \theta_{\text{nom}}) + \nabla_u f(u_k^r, \theta_{\text{nom}}).(u_k - u_k^r) + \nabla_\theta f(u_k^r, \theta_{\text{nom}}).\delta\theta^s \\ & + \vartheta(|u_k - u_k^r|^2) + \vartheta(|\delta\theta^s|^2) \end{aligned} \tag{5.37}$$

where $\vartheta(|.|^2)$ is the remainder term of the Taylor series expansion limited to the first order.

***Assumption* 5.4.** The error with respect to the first Taylor expansion, w_p, defined as:

$$w_p \triangleq \vartheta(|u_k - u_k^r|^2) + \vartheta(|\delta\theta^s|^2) \tag{5.38}$$

is assumed to be bounded as follows:

$$\exists \eta_1 \in \mathbb{R}^+, \quad \text{such that} \quad |w_p| \leq \eta_1 \tag{5.39}$$

Consider the prediction model (Taylor series expansion) for time t_{k+1}, starting from state x_k at time t_k:

$$\begin{aligned}\hat{x}_{k+1|k} &= f_p(x_k, u_k, \theta_{\text{nom}} + \delta\theta) \\ &\triangleq f(x_k, u_k^r, \theta_{\text{nom}}) + \nabla_u f(u_k^r, \theta_{\text{nom}}).(u_k - u_k^r) + \nabla_\theta f(u_k^r, \theta_{\text{nom}}).\delta\theta\end{aligned} \tag{5.40}$$

with

$$|\delta\theta| \leq |\delta\theta_{\max}| \tag{5.41}$$

with f_p the prediction model such as in (5.13) (given in this case by function g linearized as in (5.13), and using relation (4.7), with the fact that $g \equiv f$ when considering the evolution between k and $k+1$) and $\delta\theta$ the predicted parameter values mismatch.

***Assumption* 5.5.** The uncertainty on $\delta\theta \in \tilde{\Theta}$, with $\tilde{\Theta}$ a compact set containing the origin, is such that $\exists\, \eta_2 \in \mathbb{R}^+$, a modelled bound of uncertainties, so that

$$\max(|\delta\theta|, |\delta\theta^s|) \leq \eta_2 \tag{5.42}$$

Let us define $\eta \in \mathbb{R}^+$ by:

$$\eta = \max(\eta_1, \eta_2) \tag{5.43}$$

From (5.37) and (5.40), the prediction error at time index $k+1$ for $u_k = \overset{\star}{u}_k$ is given by:

$$x_{k+1} - \hat{x}_{k+1|k} = \nabla_\theta f(u_k^r, \theta_{\text{nom}}).(\delta\theta^s - \delta\theta) + w_p \tag{5.44}$$

Thanks to the triangle inequality and using the *Assumptions* 5.3 and 5.5, we obtain an upper bound on the prediction error as follows:

$$\begin{aligned}|x_{k+1} - \hat{x}_{k+1|k}| &\leq |\nabla_\theta f(u_k^r, \theta_{\text{nom}})|.|(\delta\theta^s - \delta\theta)| + |w_p| \\ &\leq |\nabla_\theta f(u_k^r, \theta_{\text{nom}})|.\,(|\delta\theta^s| + |\delta\theta|) + |w_p|\end{aligned} \tag{5.45}$$

From *Assumptions* 5.2, 5.3 and 5.5, we have:

$$\exists \alpha \in \mathbb{R}^+, \forall x, \forall u, \forall \theta, \quad |\nabla_\theta f| \leq \alpha \tag{5.46}$$

where α is an upper bound of $|\nabla_\theta f|$.
Thanks to *Assumptions* 5.4-5.5, (5.43) and (5.46), it comes

$$\begin{aligned}|x_{k+1} - \hat{x}_{k+1|k}| &\leq 2\alpha.\eta_2 + \eta_1 \\ &\leq (2\alpha + 1)\eta \triangleq \Lambda(\eta)\end{aligned} \tag{5.47}$$

where Λ is a $\mathcal{K}_\infty$-function.

5.4.2.2 Upper and lower bounds on the optimal cost

In this subsection, we would like to find bounds on the optimal cost in order to satisfy condition (7) in *Definition* 14. In the sequel, for manipulation purposes, the optimal cost function (5.32) is rewritten as follows:

$$\Pi(x_k, \overset{\star}{u}_k^{k+N_p-1}, \overset{\star}{\delta\theta}) \triangleq \sum_{t=k}^{k+N_p-1} \psi(\hat{x}_{t|k}, \overset{\star}{u}_t) + T_f(\hat{x}_{k+N_p|k}) \tag{5.48}$$

with

$$\begin{cases} \hat{x}_{t|k} = f_p(\hat{x}_{t-1|k}, \overset{\star}{u}_{t-1}, \theta_{\text{nom}} + \overset{\star}{\delta\theta}), \quad t = \overline{k+1, k+N_p} \\ \hat{x}_{k|k} = x_k \\ \psi(x_t, u_t) = \tilde{u}_t^\top v \tilde{u}_t + \tilde{y}_t^\top w \tilde{y}_t, \quad t = \overline{k, k+N_p-1} \\ T_f(\hat{x}_{k+N_p|k}) = \tilde{y}_{k+N_p}^\top w \tilde{y}_{k+N_p} \end{cases} \tag{5.49}$$

and

$$\begin{cases} \tilde{u}_t = u_t - u_t^r \\ \tilde{y}_t = H\hat{x}_t - y_t^r \end{cases} \tag{5.50}$$

The stage cost $\psi(x, u)$ is definite positive, while the terminal cost is denoted by $T_f(x) : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}_{\geq 0}$.
Without any lack of generality, the weighting matrices V and W are chosen in diagonal form

$$V = v\mathbb{I} \quad \text{and} \quad W = w\mathbb{I}$$

to simplify mathematical developments hereafter.

***Remark* 5.4.** Formulation in (5.48) and (5.32) are the same, T_f being clearly split in (5.48).

***Remark* 5.5.** The terminal stage T_f is a $\mathcal{K}_\infty$-function.

***Assumption* 5.6.** Let assume the existence of a terminal set Φ, an admissible robust positively invariant set for the system (5.36) which is controlled by the control law $u_k = \pi(x_k) \in \mathbb{U}$ s.t. the origin is in its interior. Let assume that T_f is an associated robust Lyapunov function s.t. for all $x_k \in \Phi$ and for all $\delta\theta$ satisfying (5.42), we have that:

$$\begin{aligned} &\alpha_t(|x_k|) \leq T_f(x_k) \leq \beta_t(|x_k|) + \varphi(\eta) \\ &T_f(f(x_k, u_k, \theta_{\text{nom}} + \delta\theta)) - T_f(x_k) \leq -\psi(x_k, u_k) + \chi(\eta) \end{aligned} \tag{5.51}$$

where α_t, β_t, and χ are $\mathcal{K}_\infty$-functions and φ is a $\mathcal{K}$-function.

***Lemma* 1.** Let us consider the system (5.36) and suppose that the uncertainty on θ is modelled by $|\delta\theta| \leq \gamma(|x|) + \eta$ (γ is a $\mathcal{K}-$function). Let Φ and $T_f(x)$ satisfy *Assumption* 5.6, then $\forall x \in \Phi$ we have that

$$\Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \leq T_f(x_k) + N_p\chi(\eta) \tag{5.52}$$

Proof. see *Lemma* 3 of section 5 in [86].

From (5.52) and *Assumption* 5.6, we get

$$\Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \leq \beta_t(|x_k|) + \varphi(\eta) + N_p\chi(\eta) \tag{5.53}$$

***Assumption* 5.7.** The stage cost (non-negative) is such that

$$\psi(x, u) \geq \alpha_\psi(|x|) \tag{5.54}$$

where α_ψ is a $\mathcal{K}_\infty$-function.

From (5.48) and *Assumption* 5.7, and since ψ and T_f are positive functions, then

$$\Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \geq \psi(\hat{x}_{k|k}, \overset{\star}{u}_k) \geq \alpha_\psi(|\hat{x}_{k|k}|) = \alpha_\psi(|x_k|) \tag{5.55}$$

Thanks to (5.53) and (5.55), it comes:

$$\alpha_\psi(|x_k|) \leq \Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \leq \beta_t(|x_k|) + \varphi(\eta) + N_p\chi(\eta) \tag{5.56}$$

Thus, the optimal cost is bounded as given by (5.56).

5.4.2.3 Robust stability

***Theorem* 5.2.** Consider system (5.35) and suppose that uncertainties are modelled by $|w_k| \leq \gamma(|x_k|)+\eta$. Then, the uncertain system controlled by the controller $u_k = \pi(x_k)$ is robust stable for any initial $x_0 \in X_{N_p}(\Phi)$. $X_{N_p}(\Phi)$ is the set of admissible states at time $k + N_p$. Furthermore, the optimal cost is a robust Lyapunov function.

Proof. see [86].

***Remark* 5.6.** The i-step robust stabilizable set $X_i(\Phi)$ is the set of admissible states which can be steered to the target set Φ in i steps or less by a sequence of admissible control law $\pi(x_k)$ for all possible realizations of the uncertainty. This set satisfies that $X_{i-1}(\Phi) \subseteq X_i(\Phi)$ for $i \geq 0$ with $X_0(\Phi) = \Phi$. Thanks to the invariance of the terminal set, the feasible region of the controller $X_{N_p}(\Phi)$ is a robust invariant set for the closed loop system [99].

Now, we consider $\Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta})$ as our candidate robust Lyapunov function. Then, the optimal cost function at time index $k+1$ is defined as follows:

$$\Pi(x_{k+1}, \breve{u}_{k+1}^{k+N_p}, \overset{\star}{\delta\theta}) \triangleq \sum_{t=k+1}^{k+N_p} \psi(\breve{x}_{t|k+1}, \breve{u}_t) + T_f(\breve{x}_{k+N_p+1|k+1}) \tag{5.57}$$

$$\text{with} \begin{cases} \breve{x}_{t|k+1} = f_p(\breve{x}_{t-1|k+1}, \breve{u}_{t-1}, \theta_{\text{nom}} + \overset{\star}{\delta\theta}), \quad t = \overline{k+2, k+N_p+1} \\ \breve{x}_{k+1|k+1} = x_{k+1} \end{cases} \tag{5.58}$$

where $\breve{x}_{t|k+1}$ denotes the state obtained applying the input sequence $\breve{u}_{k+1}^{t-1}$ to the prediction model with the initial condition x_{k+1}.
$\breve{u}_{k+1}^{k+N_p}$ denote an admissible solution of the optimization problem at time index $k+1$. In the proposed algorithm, it is based on the optimal solution at time index k:

$$\breve{u}_{k+1}^{k+N_p} = [\overset{\star}{u}_{k+1}^{k+N_p-1}, \overset{\star}{u}_{k+N_p-1}] \tag{5.59}$$

Notations used in this subsection are detailed in Figure 5.2.

***Assumption* 5.8.** The parameters uncertainties are assumed to be constant throughout the prediction horizon.

***Assumption* 5.9.** The function f_p is Lipschitz with respect to x with Lipschitz constant L_{fx}.

***Proposition* 5.1.** Let us define the following residual at time index l:

$$\epsilon_x(l) \triangleq \breve{x}_{l|k+1} - \hat{x}_{l|k}, \quad l = \overline{k+1, k+N_p-1} \tag{5.60}$$

Then, with *Assumption* 5.9,

$$|\epsilon_x(l)| \leq L_{fx}^{l-k-1} \Lambda(\eta) \tag{5.61}$$

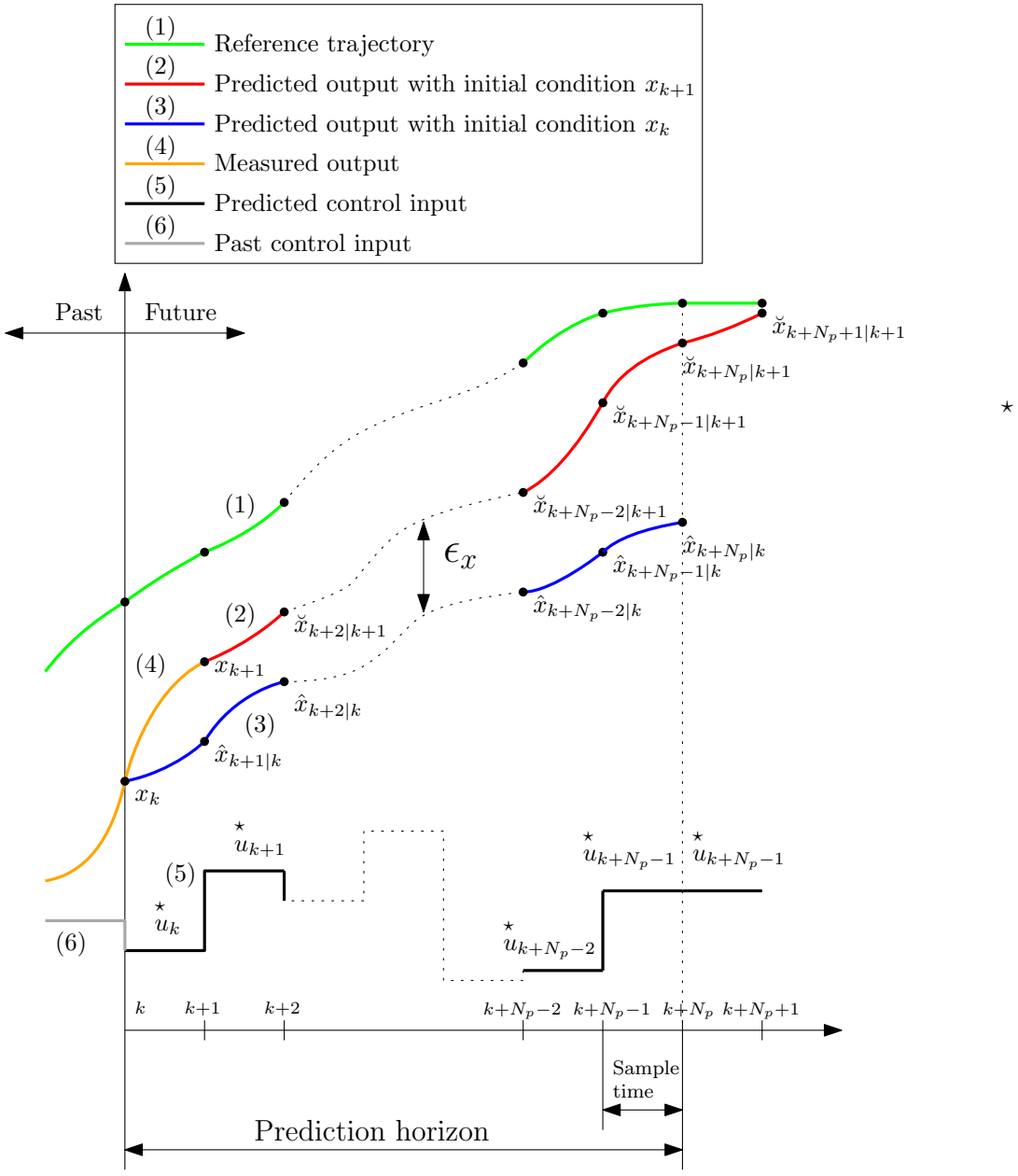

Figure 5.2: Predictions considered for the stability proof.

Proof. Using the result obtained in (5.47) and *Assumption* 5.9, we get that (reminding that $\breve{u}_l = \overset{\star}{u}_l$ for $l \in [k+1, k+N_p-1]$ and $\breve{x}_{k+1|k+1} = x_{k+1}$)

$$\begin{aligned}
|\epsilon_x(k+1)| &= |\breve{x}_{k+1|k+1} - \hat{x}_{k+1|k}| \leq \Lambda(\eta) \\
|\epsilon_x(k+2)| &= |f_p(\breve{x}_{k+1|k+1}, \breve{u}_{k+1}, \theta_{\text{nom}} + \overset{\star}{\delta\theta}) - f_p(\hat{x}_{k+1|k}, \overset{\star}{u}_{k+1}, \theta_{\text{nom}} + \overset{\star}{\delta\theta})| \\
&\leq L_{fx}|\breve{x}_{k+1|k+1} - \hat{x}_{k+1|k}| \\
&\leq L_{fx}\Lambda(\eta) \\
&\vdots \\
|\epsilon_x(l)| &\leq L_{fx}^{l-k-1}\Lambda(\eta)
\end{aligned} \tag{5.62}$$

Now, it is easy to check the result by recurrence. We assume that the proposition holds for l, let us prove that it is also the case for $l+1$ (from (5.62)):

$$\begin{aligned}
|\epsilon_x(l+1)| &= |f_p(\breve{x}_{l|k+1}, \breve{u}_l, \theta_{\text{nom}} + \overset{\star}{\delta\theta}) - f_p(\hat{x}_{l|k}, \overset{\star}{u}_l, \theta_{\text{nom}} + \overset{\star}{\delta\theta})| \\
&\leq L_{fx}|\epsilon_x(l)| \leq L_{fx}^{l-k}\Lambda(\eta)
\end{aligned} \tag{5.63}$$

Which completes the proof by recurrence.
Let define the difference $\Delta\Pi^\star$ as

$$\begin{aligned}
\Delta\Pi^\star \triangleq\ & \Pi(x_{k+1}, \breve{u}, \overset{\star}{\delta\theta}) - \Pi(x_k, \overset{\star}{u}, \overset{\star}{\delta\theta}) \\
=\ & \sum_{t=k+1}^{k+N_p-1} (\psi(\breve{x}_{t|k+1}, \breve{u}_t) - \psi(\hat{x}_{t|k}, \overset{\star}{u}_t)) \\
& + \psi(\breve{x}_{k+N_p|k+1}, \breve{u}_{k+N_p}) - \psi(x_k, \overset{\star}{u}_k) + T_f(\breve{x}_{k+N_p+1|k+1}) - T_f(\hat{x}_{k+N_p|k})
\end{aligned} \tag{5.64}$$

***Assumption* 5.10.** The stage cost ψ is Lipschitz with respect to x with Lipschitz constant $L_{\psi x}$.

Using the *Assumptions* 5.9-5.10, and from (5.62) we have that

$$|\psi(\breve{x}_{t|k+1}, \breve{u}_t) - \psi(\hat{x}_{t|k}, \overset{\star}{u}_t)| \leq L_{\psi x}|\breve{x}_{t|k+1} - \hat{x}_{t|k}| \leq L_{\psi x}L_{fx}^{t-k-1}\Lambda(\eta) \tag{5.65}$$

and

$$|\sum_{t=k+1}^{k+N_p-1} (\psi(\breve{x}_{t|k+1}, \breve{u}_t) - \psi(\hat{x}_{t|k}, \overset{\star}{u}_t))| \leq L_{\psi x}\sum_{j=0}^{N_p-2} L_{fx}^j\Lambda(\eta) \tag{5.66}$$

***Assumption* 5.11.** The terminal function T_f is Lipschitz with respect to x with Lipschitz constant L_{tx}.

Under *Assumption* 5.11, and from *Proposition* 5.1 we have that

$$|T_f(\breve{x}_{k+N_p|k+1}) - T_f(\hat{x}_{k+N_p|k})| \leq L_{tx}|\epsilon_x(k+N_p)| \leq L_{tx}L_{fx}^{N_p-1}\Lambda(\eta) \tag{5.67}$$

The last term of (5.64) is rewritten as follows:

$$\begin{aligned} T_f(\breve{x}_{k+N_p+1|k+1}) - T_f(\hat{x}_{k+N_p|k}) =& T_f(\breve{x}_{k+N_p+1|k+1}) - T_f(\breve{x}_{k+N_p|k+1}) \\ &+ T_f(\breve{x}_{k+N_p|k+1}) - T_f(\hat{x}_{k+N_p|k}) \end{aligned} \tag{5.68}$$

If the *Assumption* 5.6 holds, it comes

$$T_f(\breve{x}_{k+N_p+1|k+1}) - T_f(\breve{x}_{k+N_p|k+1}) \leq -\psi(\breve{x}_{k+N_p|k+1}, \breve{u}_{k+N_p}) + \chi(\eta) \tag{5.69}$$

Thanks to (5.67) and (5.69) and considering the fact that for any scalar $\nu \in \mathbb{R}$, $\nu \leq |\nu|$, we have that

$$T_f(\breve{x}_{k+N_p+1|k+1}) - T_f(\hat{x}_{k+N_p|k}) \leq -\psi(\breve{x}_{k+N_p|k+1}, \breve{u}_{k+N_p}) + \chi(\eta) + L_{tx}L_{fx}^{N_p-1}\Lambda(\eta) \tag{5.70}$$

By substituting the equations (5.66) and (5.70) in (5.64), and using the *Assumption* 5.7, we get that

$$\begin{aligned} \Delta\Pi^\star &\leq \sum_{t=k+1}^{k+N_p-1} (\psi(\breve{x}_{t|k+1}, \breve{u}_t) - \psi(\hat{x}_{t|k}, \overset{\star}{u}_t)) - \psi(x_k, \overset{\star}{u}_k) + \chi(\eta) + L_{tx}L_{fx}^{N_p-1}\Lambda(\eta) \\ &\leq L_{\psi x} \sum_{j=0}^{N_p-2} L_{fx}^j \Lambda(\eta) - \psi(x_k, \overset{\star}{u}_k) + \chi(\eta) + L_{tx}L_{fx}^{N_p-1}\Lambda(\eta) \\ &\leq -\alpha_\psi(|x|) + \bar{\chi}(\eta) \end{aligned} \tag{5.71}$$

with

$$\bar{\chi}(\eta) = \chi(\eta) + \left(L_{tx}L_{fx}^{N_p-1} + L_{\psi x} \sum_{j=0}^{N_p-2} L_{fx}^j \right) \Lambda(\eta) \tag{5.72}$$

where $\bar{\chi}$ is a $\mathcal{K}_\infty$-function.

According to the results obtained in (5.56) (bounds on the optimal cost) and (5.71) (evolution of the optimal cost), the optimal cost (5.48) is a robust Lyapunov function (according to *Definition 14*).
Finally, based on the *Theorem 5.2*, the system (5.36) controlled by $\pi(x) = \overset{\star}{u}_k$ is robustly stable in Φ for any uncertainty $\theta \in \Theta$ and for the considered linearized prediction model.

5.5 Unconstrained Linearized Robust Model Predictive Controller

The optimization problem (5.33)-(5.34) is solved by means of a robust regularized least squares approach in the presence of uncertain data, following an approach developed by Sayed et al. [135]. Since it is an unconstrained formulation, the control signal will be a posteriori saturated for real-time implementation. This approach is thus dedicated to the case where the constraints on the control inputs are bounds on it and there are no constraints on the state vector.

5.5.1 Robust regularized least squares problem

The different steps to solve a robust regularized least squares problem are presented below.

step 1. Solution of the OLS problem

Let us consider first the following ordinary least squares (OLS) problem:

$$\min_{z} \ ||Az - b||_W^2 \tag{5.73}$$

where

- A is a known $m \times n$ matrix;
- z is an unknown n-dimensional column vector;
- b is a known $m \times 1$ vector;
- $W \succeq 0$ is a positive-definite weighting matrix.

Assume that the matrix A is tall ($m \geq n$) and of full column rank (see *Definition* 3). Then, the OLS problem (5.73) has a unique solution, given by:

$$z^\star = (A^\top W A)^{-1} A^\top W b \tag{5.74}$$

However, this formulation can not be used for all problems (*e.g.* in the case when the matrix A is not full column rank). In order to overcome the drawbacks, a regularized least squares (RLS) can be used instead of OLS approach (5.73).
Let us consider the following RLS problem:

$$\min_{z} \ \mathcal{J}(z) \tag{5.75}$$

with

$$\mathcal{J}(z) \triangleq ||z||_V^2 + ||Az - b||_W^2 \tag{5.76}$$

$V \succ 0$ is a positive-definite weighting matrix.

The solution of the optimization problem (5.75) is given by

$$z^\star = (V + A^\top W A)^{-1} A^\top W b \tag{5.77}$$

step 2. Introduction of uncertainties in the RLS problem (5.75)

The cost function (5.76) and the optimization problem (5.75) can be modified to take into account uncertainties on A and b as follows:

$$\min_z \max_{\delta A, \delta b} \left[||z||_V^2 + ||(A + \delta A)z - (b + \delta b)||_W^2 \right] \tag{5.78}$$

where uncertainties $\delta A \in \mathbb{R}^{m \times n}$ and $\delta b \in \mathbb{R}^m$ can be structured under the following factored form:

$$\begin{cases} \delta A = C \Delta E_a & (5.79) \\ \delta b = C \Delta E_b & (5.80) \end{cases}$$

where Δ denotes an arbitrary contraction term with $||\Delta|| \leq 1$, with a known matrix $C \in \mathbb{R}^{m \times n_\xi}$ not identically null and where E_a and E_b are known quantities of appropriate dimensions.
In the sequel, the uncertainties δA and δb are replaced by a perturbation vector $\xi \in \mathbb{R}^{n_\xi}$ which is assumed to satisfy the following factored form:

$$C\xi = \delta A z - \delta b = C\Delta \left(E_a z - E_b \right) \tag{5.81}$$

Since $||\Delta|| \leq 1$, ξ is therefore constrained as follows:

$$||\xi|| \leq ||E_a z - E_b|| \triangleq \Gamma(z) \tag{5.82}$$

The nonnegative function $\Gamma(z)$ is assumed to be a known bound on the perturbation ξ and is a function of z only.
Thanks to (5.81) and (5.82), the optimization problem (5.78) can be expressed as follows:

$$\min_z \max_{||\xi|| \leq \Gamma(z)} \left[||z||_V^2 + ||Az - b + C\xi||_W^2 \right] \tag{5.83}$$

The maximization subproblem is transformed to a standard form, which will enable further to define the corresponding Lagrange dual problem:

$$\min_z \min_{||\xi|| \leq \Gamma(z)} \left[-||z||_V^2 - ||Az - b + C\xi||_W^2 \right] \tag{5.84}$$

step 3. Inclusion of the previous constraint (5.82) **in the robust RLS problem** (5.84)

The constrained subproblem on ξ is solved by considering the Lagrangian duality [23]. We define the Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^{n_\xi} \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ associated with the optimization problem (5.84) as

$$L(z, \xi, \lambda) \triangleq -||z||_V^2 - ||Az - b + C\xi||_W^2 + \lambda(||\xi||^2 - \Gamma(z)^2) \tag{5.85}$$

where λ is the Lagrange multiplier associated to the inequality constraint (5.82) on ξ. Consequently, the problem (5.84) becomes equivalent to

$$\min_z \max_{\lambda \geq 0} \min_\xi L(z, \xi, \lambda) \tag{5.86}$$

step 4. Solving the problem (5.86)

Since $L(z, \xi, \lambda)$ is a convex quadratic function of ξ, we can find an explicit solution of ξ which depends on the two variables z and λ by cancelling the gradient of the Lagrangian with respect to ξ, leading to:

$$\xi^\star(z, \lambda) = (\lambda \mathbb{I} - C^\top W C)^\dagger C^\top W (Az - b) \tag{5.87}$$

where $\mathbb{I}$ is the identity matrix with appropriate dimension.
Due to the fact that the Hessian of the Lagrangian function (5.85) with respect to ξ must be non-negative at the optimum:

$$\frac{\partial^2 L}{\partial \xi^2} = -C^\top W C + \lambda \mathbb{I} \succeq 0 \tag{5.88}$$

It turns out that the dual variable λ must satisfy the following inequality constraint

$$\lambda \geq ||C^\top W C|| \tag{5.89}$$

Thanks to the(5.87) and (5.89), problem (5.86) becomes

$$\min_z \max_{\lambda \geq ||C^\top W C||} L(z, \lambda) \tag{5.90}$$

with

$$L(z, \lambda) = -||z||_V^2 - ||Az - b||_{W(\lambda)}^2 - \lambda \Gamma(z)^2 \tag{5.91}$$

in which the modified weighting matrix $W(\lambda)$ is derived from W via:

$$W(\lambda) = W + WC(\lambda \mathbb{I} - C^\top W C)^\dagger C^\top W \tag{5.92}$$

Details concerning how to obtain (5.91) and (5.92) can be found in Appendix A.

The optimization problem (5.90) is further replaced by:

$$\min_{z} \min_{\lambda \geq ||C^\top W C||} \mathcal{J}(z,\lambda) \Longleftrightarrow \min_{\lambda \geq ||C^\top W C||} \min_{z} \mathcal{J}(z,\lambda) \tag{5.93}$$

where the cost function $\mathcal{J}(z,\lambda)$ is defined as follows

$$\mathcal{J}(z,\lambda) \triangleq ||z||_V^2 + ||Az-b||_{W(\lambda)}^2 + \lambda \Gamma(z)^2 \tag{5.94}$$

The optimal value of z for every fixed value of λ is determined by cancelling the derivative of $\mathcal{J}(z,\lambda)$ with respect to z (*i.e.* $\nabla_z \mathcal{J}(z,\lambda) = 0$).
Consequently, the minimum z must satisfy the following equation

$$\left(V + A^\top W(\lambda) A\right) z + \frac{1}{2}\lambda \nabla_z \left(\Gamma(z)^2\right) = A^\top W(\lambda) b \tag{5.95}$$

From (5.82), $\nabla_z \left(\Gamma(z)^2\right)$, the gradient of $\Gamma(z)^2$ with respect to z is given by:

$$\nabla_z \left(\Gamma(z)^2\right) = 2E_a^\top (E_a z - E_b) \tag{5.96}$$

From (5.96), the solution of the equation (5.95), which dependents on λ, is given by:

$$z(\lambda) = E(\lambda)^\dagger B(\lambda) \tag{5.97}$$

with

$$\begin{cases} E(\lambda) = V(\lambda) + A^\top W(\lambda) A & (5.98) \\ B(\lambda) = A^\top W(\lambda) b + \lambda E_a^\top E_b & (5.99) \end{cases}$$

The modified weighting matrix $V(\lambda)$ is obtained from V via:

$$V(\lambda) = V + \lambda E_a^\top E_a \tag{5.100}$$

The invertibility of $E(\lambda)$ is guaranteed by the positive definiteness of V.
By replacing $z(\lambda)$ by its optimal value given by (5.97) in the right side of (5.93), the nonnegative scalar parameter $\lambda^\star \in \mathbb{R}$ solution of (5.93) is determined from the following unidimensional minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||C^\top W C||} \left[||z(\lambda)||_V^2 + \lambda ||E_a z(\lambda) - E_b||^2 + ||Az(\lambda) - b||_{W(\lambda)}^2 \right] \tag{5.101}$$

Finally, the problem has a unique global minimum $z^\star$ given by (5.97) for $\lambda = \lambda^\star$, solution of (5.101) (*i.e.* $z^\star = z(\lambda^\star)$).
More details related to this development can be found in [135].

5.5.2 Linearized Robust Model Predictive Control

Based on this formalism, the approximated min-max optimization problem which is defined by problem (5.1) with criterion (5.32), is written in the form (5.78)-(5.82) with:

$$\min_{z} \quad \max_{||\xi|| \leq ||E_a z - E_b||} \left[||z||_V^2 + ||Az - b + C\xi||_W^2 \right] \tag{5.102}$$

and

$$\begin{cases} z = u_k^{k+N_p-1} - u_k^{r,k+N_p-1} \\ A = G_{u,k}^{k+N_p-1} \\ b = y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} \\ C = G_{\theta,k+1}^{k+N_p} \\ \Delta = \gamma, E_a = 0, E_b = -\delta\theta_{\max} \end{cases} \tag{5.103}$$

The application of (5.97-5.101) provides the solution of (5.1) with criterion (5.32) as follows:
step 1. The scalar $\lambda^\star$ is computed from the following minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p^\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(\lambda) \tag{5.104}$$

where the function $\mathbb{J}(\lambda)$ is defined by (see (5.101)):

$$\mathbb{J}(\lambda) = ||z(\lambda)||_V^2 + \lambda||\delta\theta_{\max}||^2 + ||G_{u,k}^{k+N_p-1} z(\lambda) - y_{k+1}^{r,k+N_p} + G_{\text{nom},k+1}^{k+N_p}||_{W(\lambda)}^2 \tag{5.105}$$

with (from (5.97))

$$z(\lambda) = (V + G_{u,k}^{k+N_p-1^\top} W(\lambda) G_{u,k}^{k+N_p-1})^\dagger (G_{u,k}^{k+N_p-1^\top} W(\lambda)(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p})) \tag{5.106}$$

and (from (5.92))

$$W(\lambda) = W + W G_{\theta,k+1}^{k+N_p} (\lambda \mathbb{I} - G_{\theta,k+1}^{k+N_p^\top} W G_{\theta,k+1}^{k+N_p})^\dagger G_{\theta,k+1}^{k+N_p^\top} W \tag{5.107}$$

step 2. The control sequence $\overset{\star}{u}_k^{k+N_p-1}$ is then given by (from (5.106) for $\lambda = \lambda^\star$):

$$\begin{aligned} \overset{\star}{u}_k^{k+N_p-1} =& u_k^{r,k+N_p-1} + (V + G_{u,k}^{k+N_p-1^\top} W(\lambda^\star) G_{u,k}^{k+N_p-1})^\dagger \\ & \times (G_{u,k}^{k+N_p-1^\top} W(\lambda^\star)(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p})) \end{aligned} \tag{5.108}$$

with $W(\lambda^\star)$ given by (5.107) for $\lambda = \lambda^\star$.

To summarize, the predictive controller consists in solving online an unidimensional optimization problem (5.104)-(5.105) at each sampling time, instead of solving the min-max problem (5.1)-(5.3). In the sequel, this predictive control law will be referred to as Linearized Robust Model Predictive Controller (LRMPC). The LRMPC algorithm is summarized hereafter:

LRMPC Algorithm

Inputs:
T_s: sampling time;
T_d: integration time step;
y^r, u^r: reference outputs and control inputs, respectively;
x_0: initial state vector;
θ_{nom}: nominal parameters;
$\delta\theta_{\max}$: maximum parameters uncertainties;
N_p: length of the prediction horizon;
W, V: weighting matrices on the outputs and the control inputs, respectively;

Outputs:
1. Initialisation: $k = 1$
2. Update x_k, $y_{k+1}^{r,k+N_p}$, $u_k^{r,k+N_p-1}$
3. Compute $G_{\text{nom},k}^{k+N_p-1}$, $G_{\theta,k}^{k+N_p-1}$, $G_{u,k}^{k+N_p-1}$ according to equations (5.19-5.21)
4. Optimize $\lambda^\star$ by solving the unidimensional problem (5.104)
5. Compute $\overset{\star}{u}_k^{k+N_p-1}$ according to the equation (5.108)
6. Apply $\overset{\star}{u}_k$ to the system (4.6) after saturation between $u_{\min}$ and $u_{\max}$
7. Save x_{k+1}
8. $k \longleftarrow k+1$
9. return to 2

The LRMPC problem is implemented as stated in Figure 5.3.

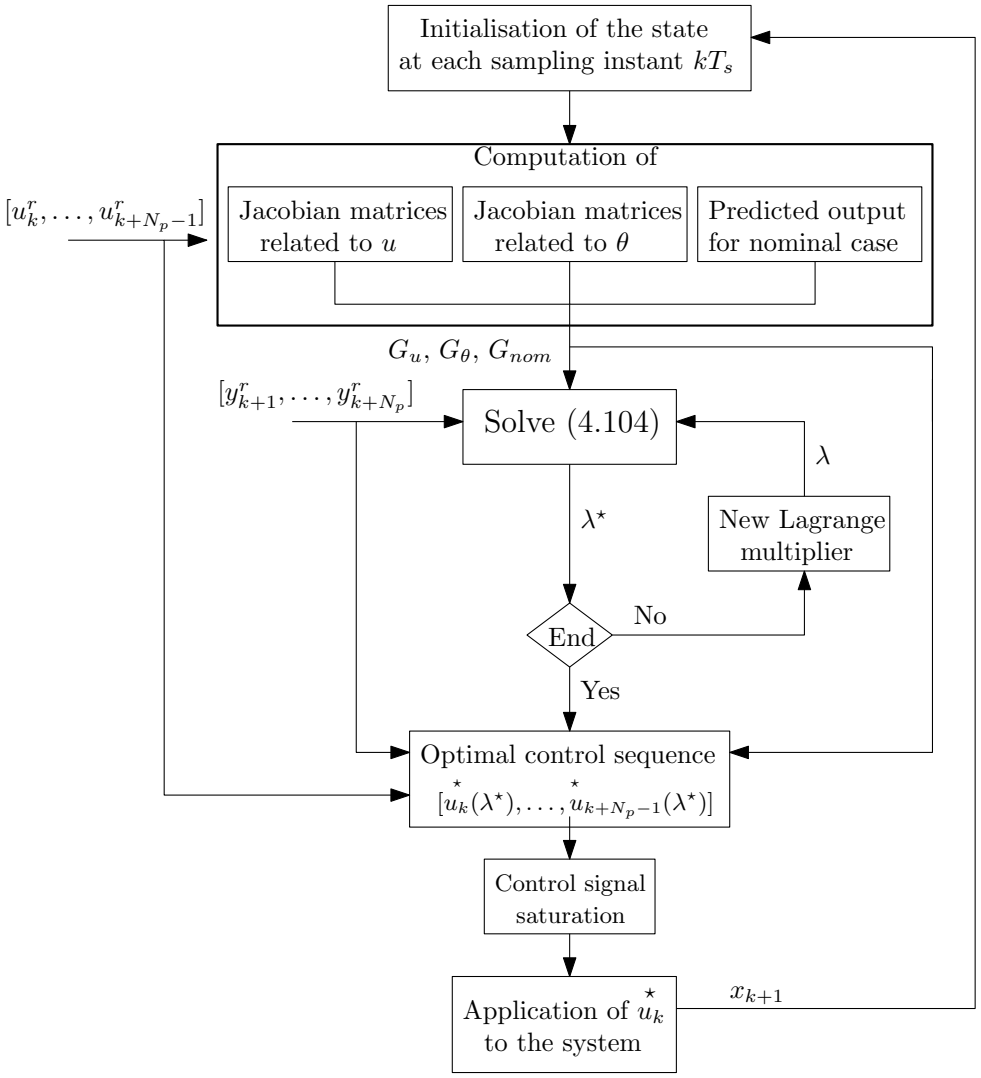

Figure 5.3: Diagram of the LRMPC algorithm.

5.6 Constrained linearized robust predictive controller

As mentionned before, the control problem that determines the control sequence has been formulated till now in an unconstrained case, with a posteriori saturation of the control signal for real-time implementation. In the sequel, inequality constraints on the control input u are taken into account in the optimization problem (*e.g.* corresponding to physical constraints on the actuators).

In this section, we will therefore consider the robust nonlinear predictive problem which is defined by (5.33)-(5.34) subject to the following inequality constraint on the optimization variable $u_k^{k+N_p-1}$:

$$u_{\min} \leq u_{k+j-1} \leq u_{\max}, \;\; j = \overline{1, N_p} \tag{5.109}$$

where $u_{\min}$ and $u_{\max}$ are the lower and upper bounds on the control signal, respectively. No constraints on the state are considered.

Based on the formalism presented in section 5.5.1, the optimization problem (5.1-5.3) with constraints (5.109) can be approached by the two player game problem (5.78). In the sequel, we extend results presented in [135] to the case of constrained robust regularized least squares. The main contribution is to solve the robust regularized optimization problem (5.78) taking into account constraints on the optimization variable z. Then, the unconstrained optimization problem defined in (5.93) with (5.94) becomes:

$$\min_{\lambda \geq ||C^\top W C||} \; \min_{\underline{z} \leq z \leq \overline{z}} \; \mathbb{J}(z, \lambda) \tag{5.110}$$

where the cost function $\mathbb{J}(z, \lambda)$ is still defined by (5.94).
From (5.82), the optimization problem (5.110) is rewritten as:

$$\min_{\lambda \geq ||C^\top W C||} \; \min_{\underline{z} \leq z \leq \overline{z}} \; \left[||z||_V^2 + \lambda ||E_a z - E_b||^2 + ||Az - b||_{W(\lambda)}^2 \right] \tag{5.111}$$

5.6.1 Bilevel optimization problem

The corresponding formulation of the optimization problem (5.111) into a bilevel optimization problem can be written as follows:

$$\begin{aligned} \lambda^\star = \arg \min_{\lambda \geq ||C^\top W C||} & \; \mathbb{J}(z(\lambda), \lambda) \\ s.t. \;\; z(\lambda) = \arg \min_{\underline{z} \leq z \leq \overline{z}} & \; \mathbb{J}(z, \lambda) \end{aligned} \tag{5.112}$$

The problem (5.112) is made of two levels of optimization problems:

- **Lower-level**
  The minimum $z(\lambda)$ is the solution of the following quadratic programming problem:

$$\begin{aligned} &\min_{z} \ \frac{1}{2} z^\top \mathcal{H} z + \mathcal{F}^\top z \\ &\text{subject to} \ \begin{bmatrix} \mathbb{I} & \mathbb{0} \\ \mathbb{0} & -\mathbb{I} \end{bmatrix} z \leq \begin{bmatrix} \overline{z} \\ -\underline{z} \end{bmatrix} \end{aligned} \tag{5.113}$$

  with

$$\begin{cases} \mathcal{H} = 2\left(V(\lambda) + A^\top W(\lambda) A\right) \\ \mathcal{F} = -2\left(A^\top W(\lambda) b + \lambda E_a^\top E_b\right) \end{cases} \tag{5.114}$$

  where $V(\lambda)$ and $W(\lambda)$ are given by (5.100) and (5.92), respectively.

- **Upper-level**
  The nonnegative scalar parameter $\lambda^\star \in \mathbb{R}^+$ is computed from the following unidimensional minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||C^\top W C||} \left[||z(\lambda)||_V^2 + \lambda ||E_a z(\lambda) - E_b||^2 + ||Az(\lambda) - b||_{W(\lambda)}^2 \right] \tag{5.115}$$

Finally, the bilevel problem (5.112) has a unique global minimum $\overset{\star}{z}$ given by (5.113) for $\lambda = \lambda^\star$ (*i.e.* $\overset{\star}{z} = z(\lambda^\star)$).

***Remark* 5.7.** Of course, if there are no constraints on z, the solution of problem (5.112) is equivalent to the one of the problem (5.101).

5.6.2 Constrained Linearized Robust Model Predictive Control

The application of (5.113)-(5.115) provides the solution of (5.1) with criterion (5.32) and inequality constraints on $u_k^{k+N_p-1}$ as follows:

Step 1. The scalar $\lambda^\star$ is computed from the following minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(z(\lambda), \lambda) \tag{5.116}$$

where the function $\mathbb{J}(z(\lambda), \lambda)$ is defined by (see (5.105)):

$$\mathbb{J}(z(\lambda), \lambda) = ||z(\lambda)||_V^2 + \lambda ||\delta\theta_{\max}||^2 + ||G_{u,k}^{k+N_p-1} z(\lambda) - y_{k+1}^{r,k+N_p} + G_{\text{nom},k+1}^{k+N_p}||_{W(\lambda)}^2 \tag{5.117}$$

and $z(\lambda)$ is given by

$$z(\lambda^\star) = \arg\min_{\underline{z} \leq z \leq \overline{z}} \left[z^\top E(\lambda) z - 2B(\lambda)^\top z \right] \tag{5.118}$$

where

$$\begin{cases} E(\lambda) = V + G_{u,k}^{k+N_p-1^\top} W(\lambda) G_{u,k}^{k+N_p-1} & (5.119) \\ B(\lambda) = G_{u,k}^{k+N_p-1^\top} W(\lambda) \left(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} \right) & (5.120) \\ \underline{z} = u_{min} \mathbb{I}_{n_u} - u_k^{r,k+N_p-1} & (5.121) \\ \overline{z} = u_{max} \mathbb{I}_{n_u} - u_k^{r,k+N_p-1} & (5.122) \end{cases}$$

with $W(\lambda)$ given in (5.107).

Step 2. The control sequence $\overset{\star}{u}_k^{k+N_p-1}$ is derived from (5.118) for $\lambda = \lambda^\star$:

$$\overset{\star}{u}_k^{k+N_p-1} = u_k^{r,k+N_p-1} + z(\lambda^\star) \tag{5.123}$$

To summarize, the predictive controller consists in solving online a bilevel optimization problem (5.112) instead of solving a min-max problem (5.1-5.3): a quadratic programming problem (5.113) in the lower level, and an unidimensional optimization problem (5.115) in the upper level. Since there are very efficient algorithms for this kind of optimization problems and that the two problems are convex, the obtained optimization problem remains more tractable than the min-max one. In the sequel, this predictive control law will be referred to as Constrained Linearized Robust Model Predictive Controller (CLRMPC). The CLRMPC algorithm is summarized hereafter.

CLRMPC Algorithm

Inputs:
T_s: sampling time;
T_d: integration time step;
y^r, u^r: reference outputs and control inputs, respectively;
x_0: initial state vector;
θ_{nom}: nominal parameters;
$\delta\theta_{\max}$: maximum parameters uncertainties;
N_p: length of the prediction horizon;
W, V: weighting matrices on the outputs and the control inputs, respectively;

Outputs:
1. Initialisation: k=1
2. Update x_k, $y_{k+1}^{r,k+N_p}$, $\overset{\star}{u}_{k-1}^{k+N_p-2}$;
3. Compute $G_{\text{nom},k}^{k+N_p-1}$, $G_{\theta,k}^{k+N_p-1}$, $G_{u,k}^{k+N_p-1}$ according to equations (5.19-5.21);
4. Solve the bilevel optimization problem (5.112);
4.1. Optimize $\lambda^\star$ by solving the unidimensional problem (5.116), by solving QP problem (5.118) for each λ;
4.2. Optimize $\overset{\star}{u}_k^{k+N_p-1}$ by solving the QP problem (5.118) for $\lambda = \lambda^\star$;
5. Apply $\overset{\star}{u}_k$ to the system (4.6);
6. Save x_{k+1};
7. $k \longleftarrow k+1$
8. return to 2

The CLRMPC problem is implemented as stated in Figure 5.4.

It should be mentionned that the CLRMPC approach can be used in the case of more complex constraints on the control input (not only bounds). In this case, the problem (5.118) becomes NLP problem with quadratic cost function and nonlinear constraints. It can be solved with an SQP algorithm for example.

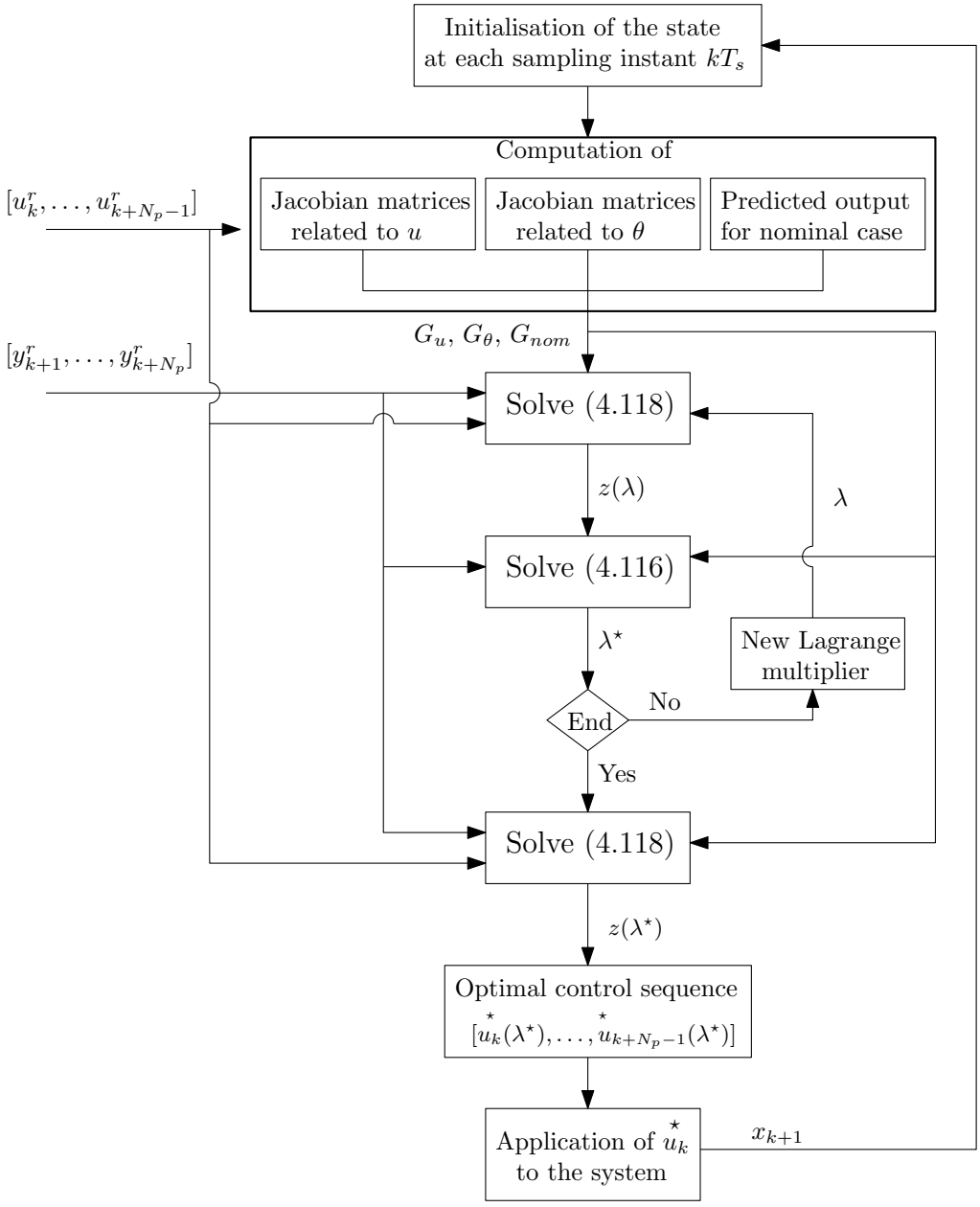

Figure 5.4: Diagram of the CLRMPC algorithm.

5.7 Numerical results and discussion

The following section illustrates the properties of both the RNMPC and the linearized formulation, using the bioprocess described in section 4.4. In fact, for this bioprocess, the rRNMPC cannot be used since the size of the parameters vector is small. This approach will be discussed in Chapter 7 on a more complex system. First, the tuning of the LRMPC is discussed in section 5.7.1. Secondly, the section 5.7.2 will compare unconstrained and constrained formulations of the LRMPC. Since the control is bounded ($0 \leq D \leq D_{\max} = 0.5\, h^{-1}$), an a posteriori saturation is applied to control inputs determined by LRMPC. Simulations have been carried out considering the uncertain parameter case discussed in the previous chapter (see section 4.4.2).

5.7.1 LRMPC tuning

In the sequel, the performances of the LRMPC under parameter uncertainties are analyzed, and more specifically the impact of its tuning parameters (T_s and N_p) on the tracking accuracy with $V = W = \mathbb{I}_{N_p}$. The same scenario of reference trajectory as in section 4.4 is considered.

First, the results as depicted in Figures 5.5-5.8 show the influence of the sampling time T_s on the concentrations of biomass and substrate, tracking error and dilution rate evolutions with a prediction horizon $N_p = 6$. The integration time step, T_d, is chosen equal to $T_s/50$.
It can be observed that the smallest sampling time provides the best tracking accuracy for the closed-loop system (Figure 5.7). In fact, T_s must be small to guarantee that the first order Taylor series expansion is accurate. It should be reminded that a compromise is required to properly select an appropriate sampling time taking into account the computation burden due to potential state estimator and/or online determination of the optimal trajectory. It appears clearly that for this example T_s=5 min allows satisfying a good trade-off between linearization accuracy and computational burden.

Secondly, the choice of the prediction horizon N_p is studied for the same conditions cited previously, with $T_s = 5$ min as shown in Figures 5.9-5.12.
The prediction horizon N_p is chosen to satisfy a compromise between the computation time and a sufficient vision of the system behaviour. In fact, increasing N_p leads to a loss of accuracy (quite small in this case) which is due to the prediction over the moving horizon using an approximated model (linearization) in the presence of model parameters uncertainties. It will nevertheless lead to a better anticipation to the future variation of reference trajectory. Hereafter, the prediction horizon is set to $N_p = 6$, which represents the best trade-off between these two criteria for this example.

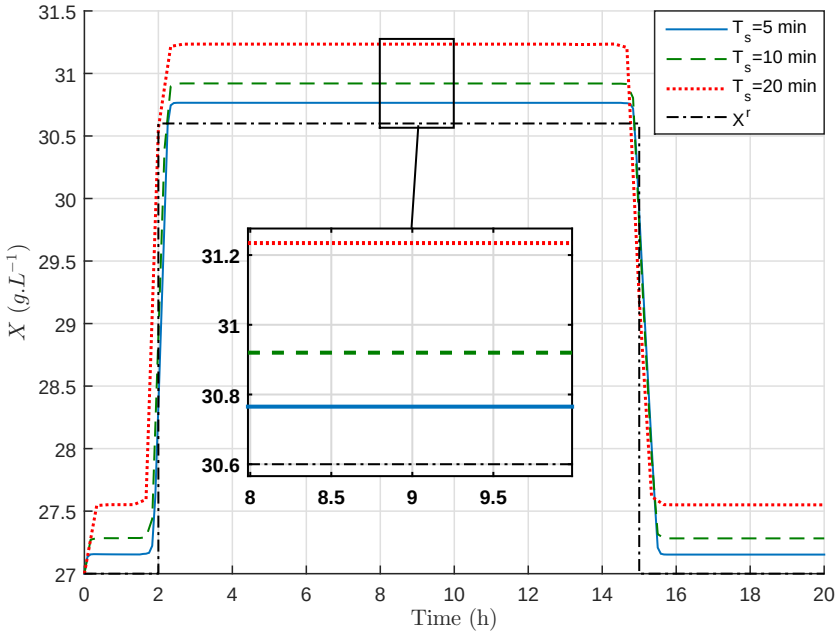

Figure 5.5: Biomass concentration evolution with time for LRMPC strategy for several values of the sampling time T_s.

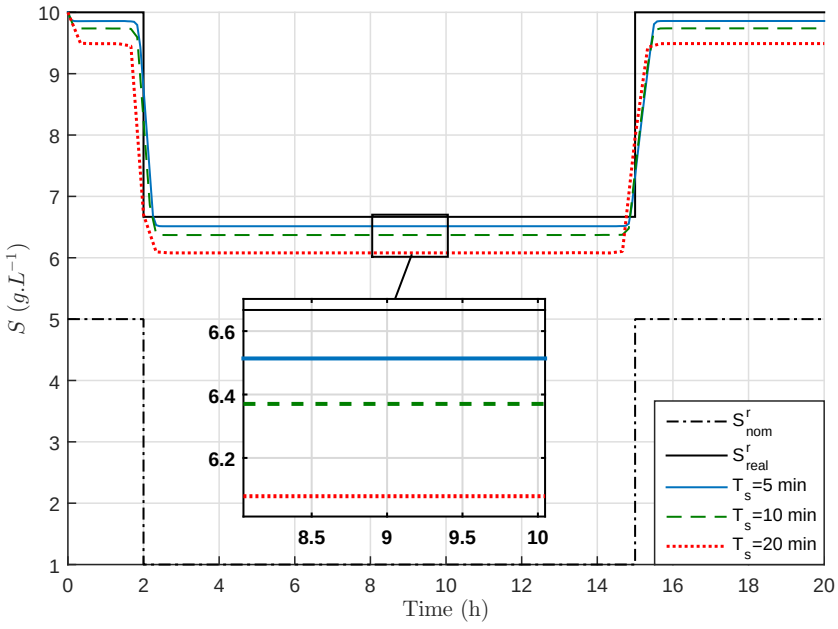

Figure 5.6: Substrate concentration evolution with time for LRMPC strategy for several values of the sampling time T_s.

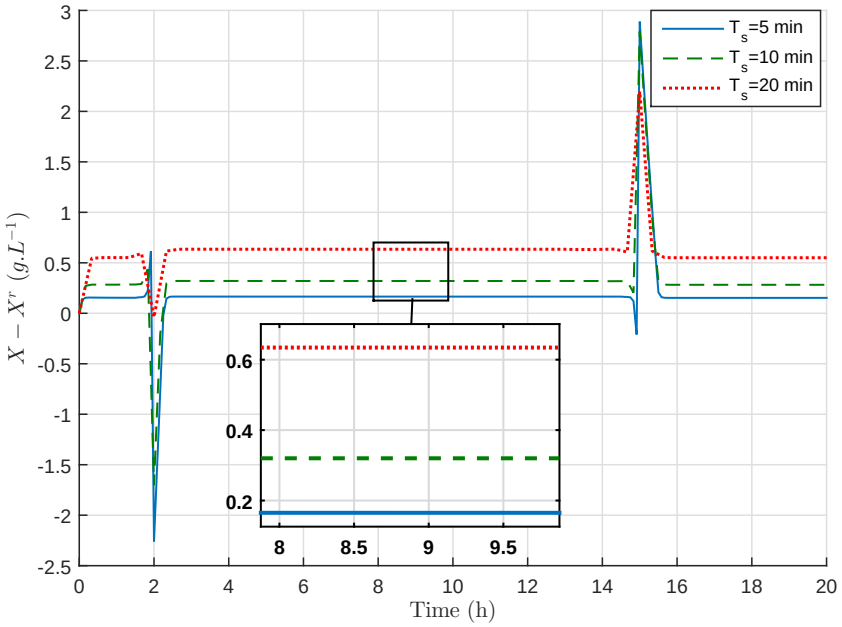

Figure 5.7: Tracking error evolution with time for LRMPC strategy for several values of the sampling time T_s.

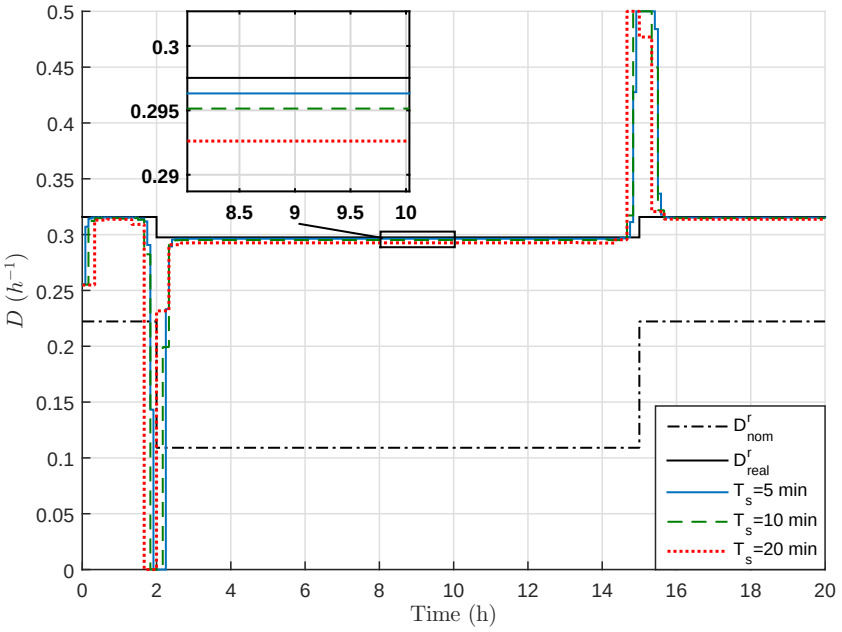

Figure 5.8: Control input evolution with time for LRMPC strategy for several values of the sampling time T_s.

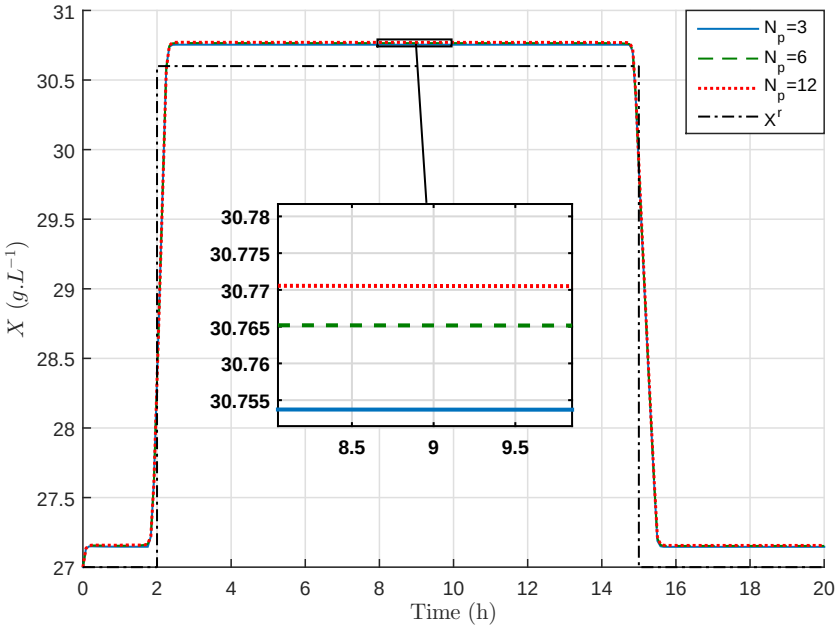

Figure 5.9: Biomass concentration evolution with time for LRMPC strategy for several values of the prediction horizon N_p.

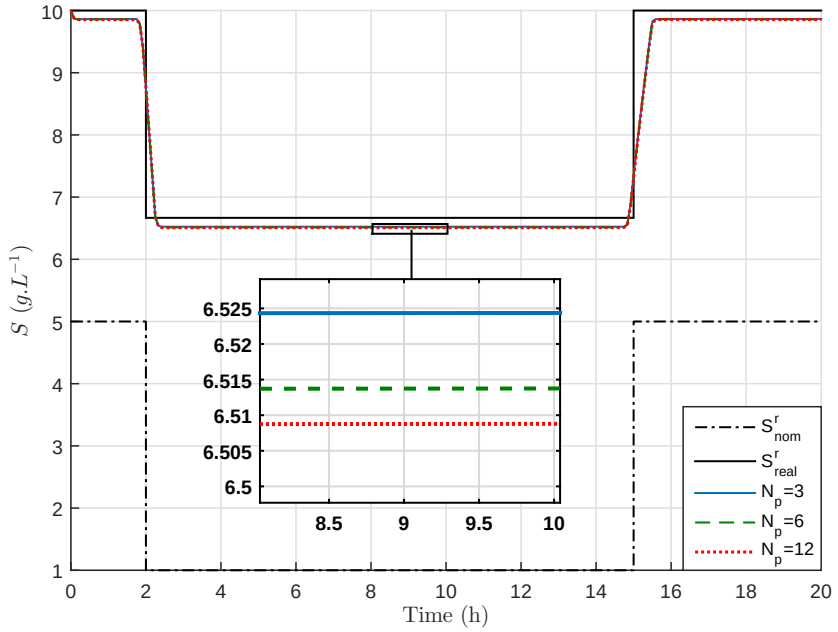

Figure 5.10: Substrate concentration evolution with time for LRMPC strategy for several values of the prediction horizon N_p.

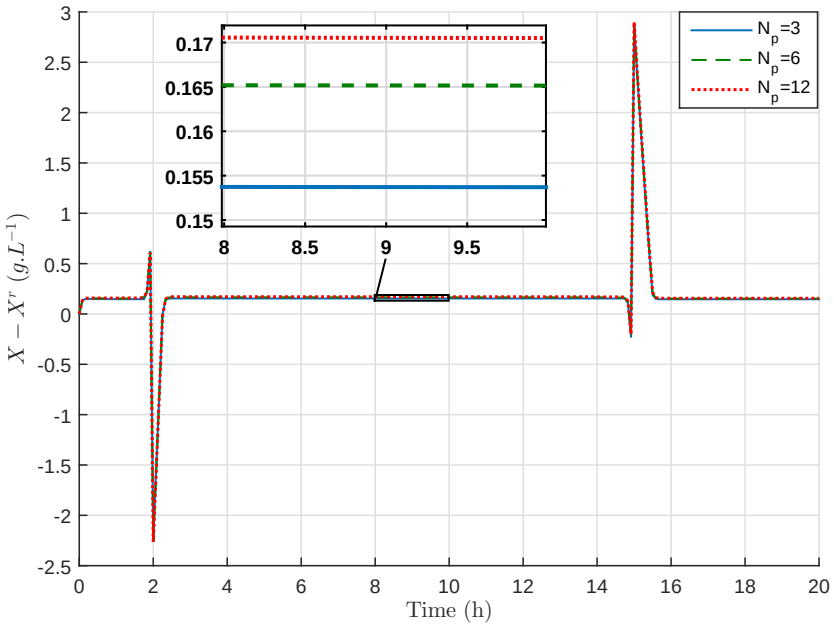

Figure 5.11: Tracking error evolution with time for LRMPC strategy for several values of the prediction horizon N_p.

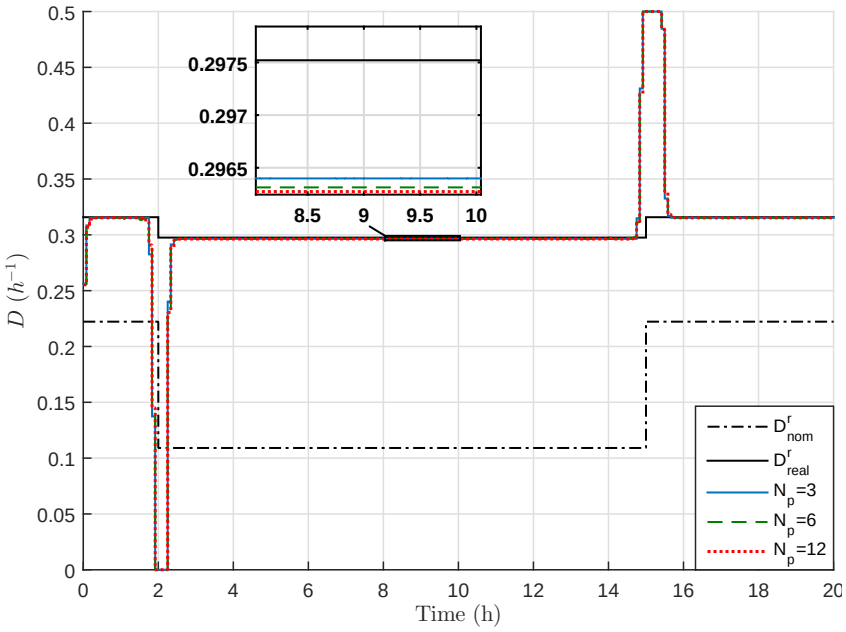

Figure 5.12: Control input evolution with time for LRMPC strategy for several values of the prediction horizon N_p.

5.7.2 Comparison of (C)LRMPC algorithms

Figures 5.13-5.15 compares both the constrained and unconstrained LRMPC (as presented in sections 5.5.2 and 5.6.2) performances obtained with the following tuning parameters: $N_p = 6$, $T_s = 5$ min, $T_d = 0.1$ min, $V = W = \mathbb{I}_{N_p}$.

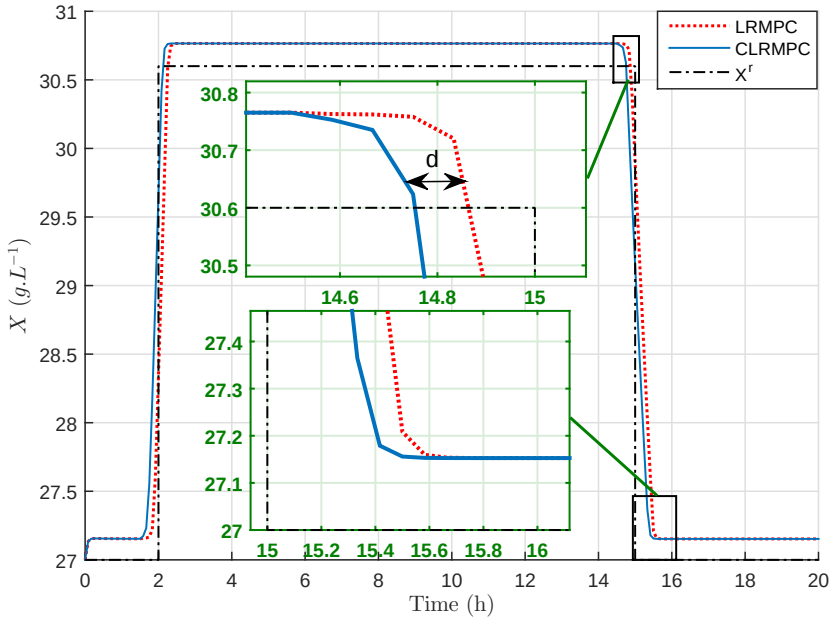

Figure 5.13: Biomass concentration evolution with time for (C)LRMPC strategies.

In the case of the LRMPC strategy, the output converges to the setpoint value with an additional time d equal to 1.5 T_s (as shown in the green box of Figure 5.13) in comparison with the CLRMPC algorithm. In fact, for CLRMPC the control is set quickly to its maximal value $D_{\max}$ while the LRMPC control input takes longer time to converge to $D_{\max}$ (see Figure 5.15). The LRMPC controller leads to suboptimal solution since it consists in solving online a scalar optimization problem (5.104-5.105) with an a posteriori saturation of the control input, whereas the constrained LRMPC law takes into account the constraints on the control inputs. On the other hand, the average computation time of the LRMPC is about fifty times less than the CLRMPC one as shown in Table 5.1. Both LRMPC and CLRMPC are equivalent outside the saturated zone (*i.e.* in case of inactive constraints). In the case of bioprocess control, the operating points are usually outside the saturated zone in particular when addressing the issue of disturbance rejection. This is a motivation for using the LRMPC strategy in the sequel.

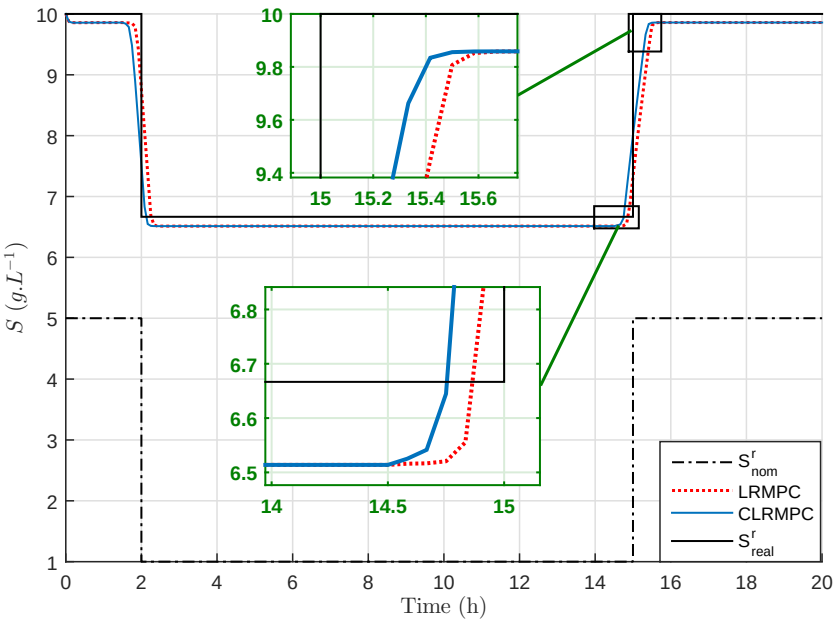

Figure 5.14: Substrate concentration evolution with time for (C)LRMPC strategies.

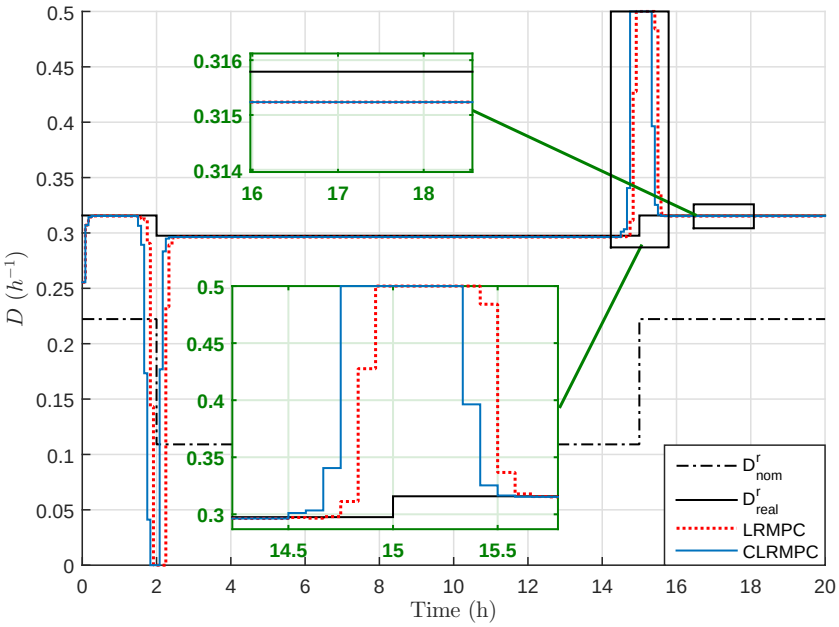

Figure 5.15: Control input evolution with time for (C)LRMPC strategies.

Table 5.1: Comparison of (C)LRMPC algorithms in terms of computation time at each sampling time (mismatched parameters).

	Computation time (s)		
Perf. indices / Algo.	min	mean	max
CLRMPC	0.23	0.48	1.7
LRMPC	$< 10^{-5}$	0.01	0.1

***Remark* 5.8.** The same phenomenon can be observed for a setpoint change in falling edge as shown in Figure 5.15.

5.7.3 Comparison of predictive controllers

In this section, three predictive control laws will be tested (Figures 5.16-5.18): a classical Nonlinear Model Predictive Control (denoted as NMPC), a robust one using criterion (5.1) (denoted as RNMPC) and the proposed one (LRMPC). The tuning parameters are the same for all strategies ($N_p = 6$, $T_s = 5$ min, $T_d = 0.1$ min and $V = W = \mathbb{I}_{N_p}$). Both RNMPC and LRMPC are compared to the NMPC in order to quantify the impact of tracking error due to the parametric uncertainty. It can be noticed the anticipation behavior to a setpoint change (Figure 5.16), due to the prediction of the output future evolution over the moving horizon and the knowledge of the setpoint trajectory in the future. The obtained results show that both RNMPC and LRMPC have better performances than NMPC under parameter uncertainties. In the NMPC law, the output is not able to track the specified setpoint in the presence of parameters uncertainties, due to the fact that the mismatch between the system and the model is not considered during the prediction step inside the minimization. The robust formulation tackles this drawback. In addition, the obtained results show that RNMPC has better performances than the LRMPC controller under parameter uncertainties in term of tracking accuracy. In the case of LRMPC, the static error is due to the approximation of the model through linearization. On the other hand, the LRMPC offers a very significant computational load reduction in comparison with the RNMPC as shown in Table 5.2. In fact, this can be explained by the fact that RNMPC is an optimization problem of dimension $N_p \times n_u \times n_\theta$ while LRMPC is a unidimensional optimization problem. Consequently, when considering a more complex model with a greater number of state variables and parameters, the computation time increases quickly in the RNMPC strategy.

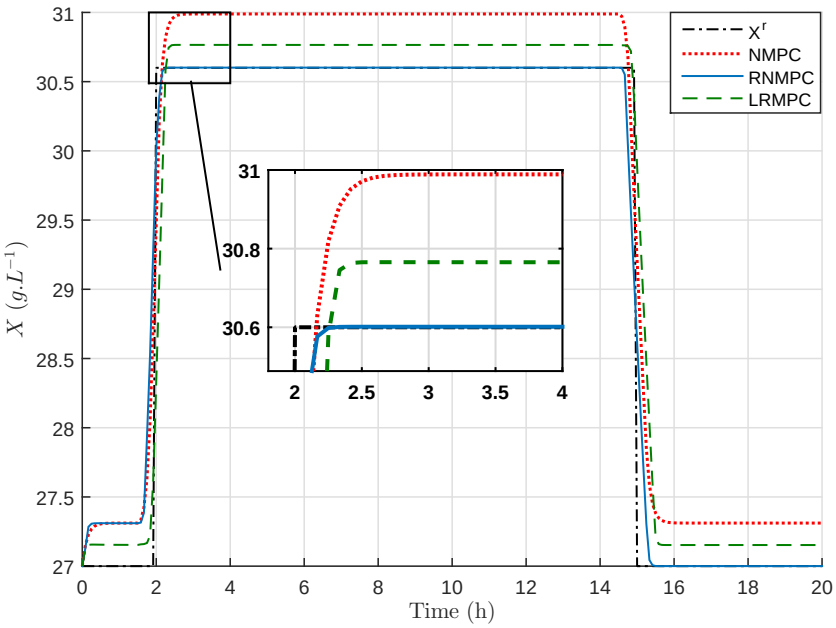

Figure 5.16: Biomass concentration evolution with time for NMPC, RNMPC and LRMPC strategies.

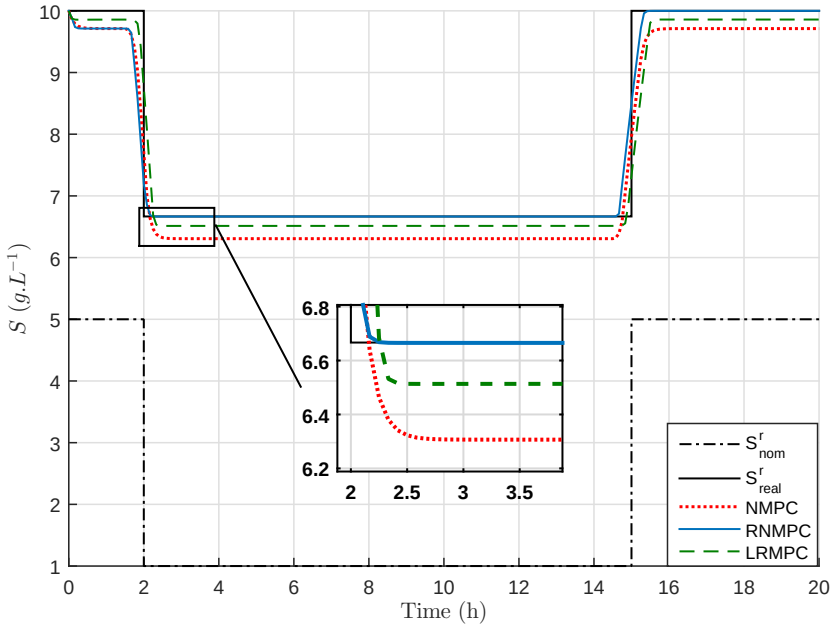

Figure 5.17: Substrate concentration evolution with time for NMPC, RNMPC and LRMPC strategies.

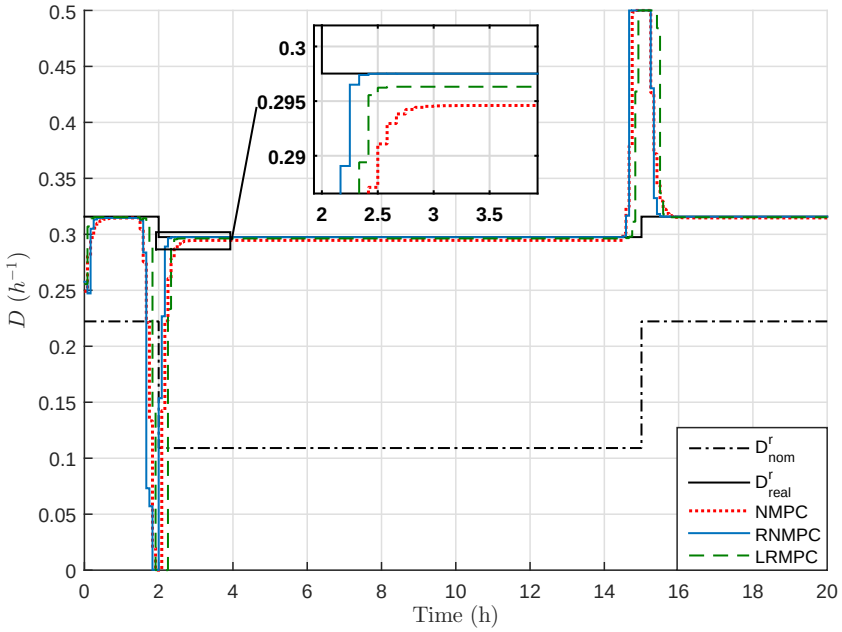

Figure 5.18: Control input evolution with time for NMPC, RNMPC and LRMPC strategies.

Table 5.2: Comparison of the proposed algorithms in terms of computation time at each sampling time (mismatched parameters).

	Computation time (s)		
Perf. indices / Algo.	min	mean	max
NMPC	$< 10^{-5}$	0.02	0.32
RNMPC	0.06	0.31	6.77
LRMPC	$< 10^{-5}$	0.01	0.1

Therefore LRMPC appears to be a good compromise between calculation time and accuracy. Moreover, a possible improvement to reduce the static error which is due to the approximation of the model through linearization, will be presented in the next chapter section 6.3.

***Remark* 5.9.** From Tables 5.1-5.2, it appears that the mean computation time for CLRMPC at each sampling time when considering the worst uncertain parameter case is greater than in the RNMPC one. This result is mainly due to the fact that the RNMPC considers the worst case in the problem formulation (pessimistic approach) which is the case for the consid-

ered model mismatch. This leads to a reduction of the mean computation time to solve the problem. In addition, the max subproblem of RNMPC is initialized from one step to another with the optimal parameters values of the previous optimization (see *Remark* 5.1), thus saving time, which is not the case of CLRMPC due to a different structure of the problem. On the contrary, in the nominal case (without parameters uncertainties) the mean computation time of CLRMPC at each time instant is less than the RNMPC one as shown in Table 5.3.

Table 5.3: Comparison of the proposed algorithms in terms of computation time at each sampling time (nominal parameters).

	Computation time (s)		
Perf. indices / Algo.	min	mean	max
CLRMPC	0.2	0.47	0.62
RNMPC	0.04	1.55	2.92

5.8 Concluding remarks

In this chapter, the robust MPC was investigated. Considering a system with parameters that are within a given confidence interval, the robust NMPC is designed in order to take into account these parameters uncertainties. It leads to a min-max optimization problem where the optimal control sequence is determined so that the maximum deviation for all trajectories overall possible data scenarii is minimized. To reduce the computation burden, we propose two alternative approaches to solve the min-max problem. In the first one, the computation time is reduced by considering only the most influencial parameters (so-called reduced RNMPC). It uses the sensitivity analysis of the model with respect to its parameters to determine the parameters to be considered in the min-max problem. In the second approach, we propose to linearize the predicted trajectory to turn the original optimization problem into a more tractable one. The derived optimization problem can be an unidimensionnal optimization problem (LRMPC) by applying the saturation on the control input a posteriori. Whereas taking into account the inequality constraints on the optimization variable in the design controller leads to a bilevel problem (CLRMPC), with a scalar optimization problem in the upper level, and a quadratic programming one in the lower level. The main advantage of LRMPC is to be computationally tractable in calculating the optimal control compared to both the bilevel problem and a classical

min-max robust approach, which makes it suitable for online implementation (even if in our case, all calculation times remain much smaller than the sampling period, this should not be the same for a more complex system, with more uncertain parameters). Several simulations were performed in order to compare the LRMPC strategy to the classical RNMPC law in the case of model parameters uncertainties to control bacteria growth in a bioreactor. The LRMPC was shown to ensure a good trade-off between computational load and tracking trajectory accuracy. However, there are several issues that deserve further investigation (e.g. non-zero tracking error). Some improvements of this control law are detailed in the next chapter.

Chapter 6

Some improvements of LRMPC

6.1 Introduction

In the previous chapter, in the LRMPC strategy, the min-max optimization problem was approached by a more tractable formulation, derived from the linearization of the system around nominal values of the parameters and reference control inputs. In this chapter, some improvements are proposed for the LRMPC controller.

A first proposed idea consists in linearizing the model around the nominal parameter values and the optimal control sequence obtained at the previous iteration instead of a reference control sequence determined at the steady state behavior. This modification is motivated by the fact that the optimal control sequence obtained at the previous iteration is non-model-based, and thus the controller robustness with respect to model uncertainties is increased. As in the case of LRMPC presented in Chapter 5, the derived optimization problem is here a bilevel one or a scalar optimization problem depending on the considered constraints. This controller will be referred as incremental LRMPC.

Furthermore, as highlighted in section 5.7.1, the LRMPC accuracy can be an issue (depending on the choice of the sampling period) as it deals with a first-order approximation of nonlinear functions. To go further with model uncertainties and linearization drawbacks, the idea here is to develop a hierarchical control scheme. The proposed control strategy combines a robust model predictive control law with a Proportional Integral (PI) law or Integral Sliding Mode (ISM) controller regulator. The predictive controller guarantees the tracking of the reference trajectory, whereas the other regulator (PI or ISM) ensures cancellation of any residual error.

The chapter is organized as follows. The section 6.2 deals with the de-

sign of the incremental LRMPC. Section 6.3 presents the hierarchical control scheme that combines a robust predictive law with an auxiliary controller. Performances of the two controllers mentionned above will be illustrated on the illustrative example studied in the previous chapters. Finally, conclusions are summarized in section 6.4.

6.2 Variant of LRMPC

This section is motivated by the problem of designing a predictive controller in the presence of bounded uncertainties and inputs. In comparison with the previous chapter, the robust cost that will be minimized is expressed as a quadratic function measuring the control efforts instead of the difference between the control inputs and their reference values which are model-based. The idea is to use a model linearization along the nominal parameters value and the optimal control sequence obtained at the previous iteration. In addition to be more robust to errors in the determination of the reference trajectory than the previous approach, it presents the advantage to lead to a solution less sensitive to noise measurements, due to the fact that the control increments are explicitly considered in the cost function [141].

6.2.1 Problem formulation

The control sequence that minimizes a worst case cost function is derived from the following optimization problem (at time index k) instead of the optimization problem (5.1):

$$\delta u_k^{\star k+N_p-1} = \arg \min_{\delta u_k^{k+N_p-1} \in \mathcal{U}^\delta} \max_{\delta\theta \in \Theta^\delta} \Pi_\delta(x_k, \delta u_k^{k+N_p-1}, \delta\theta) \tag{6.1}$$

Subject to

$$\hat{x}_{k+j} = g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta = \theta_{\text{nom}} + \delta\theta), \;\; j = \overline{1, N_p} \tag{6.2}$$

where the robust cost function is defined as

$$\Pi_\delta(x_k, \delta u_k^{k+N_p-1}, \delta\theta) \triangleq ||\delta u_k^{k+N_p-1}||_V^2 + ||\hat{y}_k^{k+N_p} - y_k^{r,k+N_p}||_W^2 \tag{6.3}$$

The optimization variables $\delta u_k^{k+N_p-1}$ are defined as the control increments as follows:

$$\delta u_k^{k+N_p-1} = \begin{bmatrix} u_k - \overset{\star}{u}_{k-1} \\ \vdots \\ u_{k+j} - u_{k+j-1} \\ \vdots \\ u_{k+N_p-1} - u_{k+N_p-2} \end{bmatrix} \triangleq \begin{bmatrix} \delta u_k \\ \vdots \\ \delta u_{k+j} \\ \vdots \\ \delta u_{k+N_p-1} \end{bmatrix} \tag{6.4}$$

with $\overset{\star}{u}_{k-1}$ the control input applied at time index $k-1$ (*i.e.* solution of (6.1) at time index $k-1$).
As in Chapter 5, the predicted outputs over the receding horizon are given by:

$$\hat{y}_{k+1}^{k+N_p} = \begin{bmatrix} Hg(t_k, t_{k+1}, x_k, u_k, \theta) \\ \vdots \\ Hg(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta) \\ \vdots \\ Hg(t_k, t_{k+N_p}, x_k, u_k^{k+N_p-1}, \theta) \end{bmatrix} \tag{6.5}$$

and $y_k^{r,k+N_p} = \begin{bmatrix} y_k^r \\ \vdots \\ y_{k+N_p}^r \end{bmatrix}$ the setpoint values.
$\mathcal{U}^\delta$ and Θ^δ are compact sets that contains the origin, defined as follows:

$$\begin{cases} \Theta^\delta = [-\delta\theta_{\max}, \delta\theta_{\max}] \text{ with } \delta\theta_{\max} = (\theta^+ - \theta^-)/2 & (6.6) \\ \mathcal{U}^\delta = [\underline{\delta u}, \overline{\delta u}] & (6.7) \end{cases}$$

$V \succeq 0$ and $W \succ 0$ are tuning weighting matrices.

The predicted state for time t_{k+j}, starting from state at t_k, is linearized around the trajectory given for the nominal parameters θ_{nom} and a control sequence $\bar{u}_k^{k+N_p-1}$ defined as the optimal control sequence of the optimization problem (6.1) obtained at the previous sampling time (at time index $k-1$):

$$\bar{u}_j = \overset{\star}{u}_{j-1}, \quad j = \overline{k, k+N_p-1} \tag{6.8}$$

As for LRMPC, a first order Taylor series (local) expansion of (4.5) for $j = \overline{1, N_p}$ is used:

$$\hat{x}_{k+j} \approx \hat{x}_{\text{nom},k+j} + \nabla_\theta g(t_{k+j})\delta\theta + \nabla_u g(t_{k+j}) \left(u_k^{k+j-1} - \bar{u}_k^{k+j-1}\right) \tag{6.9}$$

with

$$\begin{cases} \hat{x}_{\text{nom},k+j} = g(t_k, t_{k+j}, x_k, \bar{u}_k^{k+j-1}, \theta_{\text{nom}}) & (6.10) \\ \nabla_\theta g(t_{k+j}) = \left.\dfrac{\partial g(t_k, t_{k+j}, x_k, u_k^{k+j-1}, \theta)}{\partial \theta}\right|_{\substack{u_k^{k+j-1} = \bar{u}_k^{k+j-1} \\ \theta = \theta_{\text{nom}}}} & (6.11) \\ \nabla_u g(t_{k+j}) = \left.\dfrac{\partial g(t_k, t_{k+i}, x_k, u_k^{k+j-1}, \theta)}{\partial u_k^{k+j-1}}\right|_{\substack{u_k^{k+j-1} = \bar{u}_k^{k+j-1} \\ \theta = \theta_{\text{nom}}}} & (6.12) \end{cases}$$

The dynamics of sensitivity function with respect to θ and u, defined in (6.11-6.12), can be computed for time $t \in [t_k, t_{k+N_p}]$ as detailed in section 5.4.1. From (6.5) and (6.9), it comes:

$$\begin{cases} \hat{x}_{k+1} \approx \hat{x}_{\text{nom},k+1} + \nabla_\theta g(t_{k+1})\delta\theta + \nabla_u g(t_{k+1})(u_k - \bar{u}_k) \\ \vdots \\ \hat{x}_{k+j} \approx \hat{x}_{\text{nom},k+j} + \nabla_\theta g(t_{k+j})\delta\theta + \nabla_u g(t_{k+j})(u_k^{k+j-1} - \bar{u}_k^{k+j-1}) \\ \vdots \\ \hat{x}_{k+N_p} \approx \hat{x}_{\text{nom},k+N_p} + \nabla_\theta g(t_{k+N_p})\delta\theta + \nabla_u g(t_{k+N_p})(u_k^{k+N_p-1} - \bar{u}_k^{k+N_p-1}) \end{cases} \quad (6.13)$$

Using (6.4) and (6.8), the terms $u_{k+j} - \bar{u}_{k+j}$, $j = \overline{1, N_p - 1}$ are written as functions of Δu_j, $j = \overline{1, N_p - 1}$ as follows:

$$\begin{cases} u_k - \bar{u}_k = u_k - \overset{\star}{u}_{k-1} = \delta u_k \\ u_{k+1} - \bar{u}_{k+1} = \underbrace{u_{k+1} \overbrace{- u_k}^{0} + u_k}_{\delta u_{k+1}} \underbrace{- \overset{\star}{u}_{k-1}}_{\delta u_k} + \overset{\star}{u}_{k-1} - \overset{\star}{u}_k = \delta u_{k+1} + \delta u_k + \overset{\star}{u}_{k-1} - \overset{\star}{u}_k \\ \vdots \\ u_{k+j} - \bar{u}_{k+j} = \delta u_{k+j} + \ldots + \delta u_k + \overset{\star}{u}_{k-1} - \overset{\star}{u}_{k+j-1} \\ \vdots \\ u_{k+N_p-1} - \bar{u}_{k+N_p-1} = \delta u_{k+N_p-1} + \ldots + \delta u_k + \overset{\star}{u}_{k-1} - \overset{\star}{u}_{k+N_p-2} \end{cases} \quad (6.14)$$

And thus:

$$u_k^{k+N_p-1} - \bar{u}_k^{k+N_p-1} = \begin{bmatrix} \delta u_k \\ \delta u_{k+1} + \delta u_k + \overset{\star}{u}_{k-1} - \overset{\star}{u}_k \\ \vdots \\ \delta u_{k+N_p-1} + \ldots + \delta u_k + \overset{\star}{u}_{k-1} - \overset{\star}{u}_{k+N_p-2} \end{bmatrix} \quad (6.15)$$

Finally, it comes

$$u_k^{k+N_p-1} - \bar{u}_k^{k+N_p-1} = \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k-1} \end{bmatrix} - \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k+N_p-2} \end{bmatrix} + \begin{bmatrix} \mathbb{I}_{n_u} & \mathbb{0} & \dots & \mathbb{0} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \mathbb{0} \\ \mathbb{I}_{n_u} & \dots & \dots & \mathbb{I}_{n_u} \end{bmatrix} \delta u_k^{k+N_p-1}$$
$$\triangleq \Xi_k^{k+N_p-1} + T_{N_p} \delta u_k^{k+N_p-1} \tag{6.16}$$

where

$$\Xi_k^{k+N_p-1} = \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k-1} \end{bmatrix} - \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k+N_p-2} \end{bmatrix} \tag{6.17}$$

and

$$T_{N_p} \in \mathbb{R}^{N_p \times N_p} \text{: unitary lower triangular matrix} \tag{6.18}$$

From (4.11), (6.9) and (6.16), the predicted ouputs $\hat{y}_{k+1}^{k+N_p}$ over the moving horizon are expressed as follows:

$$\hat{y}_{k+1}^{k+N_p} = G_{\text{nom},k+1}^{k+N_p} + G_{\theta,k+1}^{k+N_p} \delta\theta + G_{u,k}^{k+N_p-1} (\Xi_k^{k+N_p-1} + T_{N_p} \delta u_k^{k+N_p-1}) \tag{6.19}$$

with $G_{\text{nom},k+1}^{k+N_p}$, $G_{\theta,k+1}^{k+N_p}$ and $G_{u,k}^{k+N_p-1}$ defined as in section 5.4.1 (see (5.25)). Consequently, the linearized prediction outputs have similar structure as in the LRMPC (5.25), only the last term differs. This controller will be referred to as incremental LRMPC (LRMPC-δu).

6.2.2 Stability analysis

In the previous chapter, the stability analysis has been addressed for the closed-loop with the LRMPC law. The same methodology could be used for the LRMPC-δu under the same assumptions as in section 5.4.2.

The difference is that the model is linearized around θ_{nom} and $\bar{u}$ instead of θ_{nom} and u^r.

Consequently, the bound on the prediction error given by (5.47) as well as upper and lower bounds on the optimal cost given by (5.56) as still valid.

For the robust stability (as in section 5.4.2.3), the admissible control input is still chosen as:

$$\breve{u}_{k+1}^{k+N_p} = [\overset{\star}{u}_{k+1}^{k+N_p-1}, \overset{\star}{u}_{k+N_p-1}] \tag{6.20}$$

and the relation (5.71) is still valid.

Consequently, considering the control sequence $\bar{u}$ instead of u^r do not change the stability of the controller (which is in this case mainly linked to the considered prediction model).

6.2.3 Derivation of the control law

6.2.3.1 Problem formulation

The constrained min-max optimization problem (6.1)-(6.3) is approached by a robust RLS problem (6.21)-(6.22) when applying (5.29) and (6.19) in the presence of uncertain data, as follows:

$$\min_{\underline{z}\leq z\leq \overline{z}} \max_{||\xi||\leq||E_a z-E_b||} ||z||_V^2 + ||Az - b + C\xi||_W^2 \tag{6.21}$$

$$\text{with} \begin{cases} z = \delta u_k^{k+N_p-1} \\ A = G_{u,k}^{k+N_p-1} T_{N_p} \\ b = y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} - G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1} \\ C = G_{\theta,k+1}^{k+N_p} \\ \Delta = \gamma, E_a = 0, E_b = -\delta\theta_{\max} \\ \underline{z} = \underline{\delta u}, \overline{z} = \overline{\delta u} \end{cases} \tag{6.22}$$

where the perturbation vector ξ is assumed to satisfy the following form:

$$\xi = \Delta(E_a z - E_b) \tag{6.23}$$

with

$$||\Delta|| \leq 1 \tag{6.24}$$

In this case, only bounds on the control input are considered (but the control strategy can be generalized to a more general constraints, *i.e.* $z \in \mathbb{Z}_c$).

The obtained optimization problem (6.21)-(6.22) is similar to the one obtained when considering reference control inputs in the cost function (see (5.103)) if the correspondances summarized in Table 6.1 are considered.

6.2.3.2 Bilevel optimization problem

The constrained optimization problem (6.21)-(6.22) is solved by considering the Lagrangian dual problem of the maximization subproblem (i.e. the maximization of the error over all possible values of model parameters), following a similar approach as in section 5.6.2. Indeed, problem (5.102) has the same structure as problem (6.21) as mentionned previously.
The solution of (6.21) with (6.22) is then given by:

Lower-level: The optimal value of $z = \delta u_k^{k+N_p-1}$ is derived from the following QP problem:

$$z(\lambda) = \arg\min_{\underline{z}\leq z\leq \overline{z}} z^\top E(\lambda) z - 2B(\lambda)^\top z \tag{6.25}$$

Table 6.1: Comparison of LRMPC and LRMPC-δu algorithms in term of parameters of the optimization problem.

	LRMPC	LRMPC-δu
z	$u_k^{k+N_p-1} - u_k^{r,k+N_p-1}$	$\delta u_k^{k+N_p-1}$
A	$G_{u,k}^{k+N_p-1}$	$G_{u,k}^{k+N_p-1} T_{N_p}$
b	$y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p}$	$y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} - G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1}$
C	$G_{\theta,k+1}^{k+N_p}$	$G_{\theta,k+1}^{k+N_p}$
Δ	γ	γ
E_a	0	0
E_b	$-\delta\theta_{\max}$	$-\delta\theta_{\max}$

where

$$E(\lambda) = V + T_{N_p}^\top {G_{u,k}^{k+N_p-1}}^\top W(\lambda) G_{u,k}^{k+N_p-1} T_{N_p} \tag{6.26}$$

and

$$B(\lambda) = T_{N_p}^\top {G_{u,k}^{k+N_p-1}}^\top W(\lambda) \left(y_{k+1}^{r,k+N_p} - G_{\text{nom},k+1}^{k+N_p} - G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1} \right) \tag{6.27}$$

Upper-level: The scalar $\lambda^\star$ is computed from the following minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(z(\lambda), \lambda) \tag{6.28}$$

where the function $\mathbb{J}(z(\lambda), \lambda)$ is defined by (see (5.105)):

$$\begin{aligned} \mathbb{J}(z(\lambda), \lambda) =& ||z(\lambda)||_V^2 + \lambda ||\delta\theta_{\max}||^2 + \\ & ||G_{u,k}^{k+N_p-1} T_{N_p} z(\lambda) - y_{k+1}^{r,k+N_p} + G_{\text{nom},k+1}^{k+N_p} + G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1}||_{W(\lambda)}^2 \end{aligned} \tag{6.29}$$

with $W(\lambda)$ given by (5.107).
The control sequence $\overset{\star}{u}_k^{k+N_p-1}$ is then given by:

$$\overset{\star}{u}_k^{k+N_p-1} = \begin{bmatrix} \overset{\star}{u}_{k-1} \\ \vdots \\ \overset{\star}{u}_{k-1} \end{bmatrix} + T_{N_p} \overset{\star}{z} \tag{6.30}$$

where $\overset{\star}{z}$ is obtained from (6.25) for $\lambda = \lambda^\star$ (*i.e.* $\overset{\star}{z} = z(\lambda^\star)$).

***Remark* 6.1.** The same methodology could be applied for the unconstrained problem. Then, the resulting optimization problem will be an unidimensional problem as presented in section 5.5.2:
step 1. The scalar $\lambda^\star$ is computed from the following minimization problem:

$$\lambda^\star = \arg \min_{\lambda \geq ||G_{\theta,k+1}^{k+N_p\top} W G_{\theta,k+1}^{k+N_p}||} \mathbb{J}(z(\lambda), \lambda) \tag{6.31}$$

where $\mathbb{J}(z(\lambda), \lambda)$ and $W(\lambda)$ are given by (6.29) and (5.107), respectively, with (from (5.97))

$$\begin{aligned} z(\lambda) = &\left(V + T_{N_p}^\top G_{u,k}^{k+N_p-1\top} W(\lambda) G_{u,k}^{k+N_p-1} T_{N_p}\right)^\dagger T_{N_p}^\top G_{u,k}^{k+N_p-1\top} W(\lambda) \\ &\times \left(\bar{y}_{k+1}^{k+N_p} - G_{\text{nom},k+1}^{k+N_p} - G_{u,k}^{k+N_p-1} \Xi_k^{k+N_p-1}\right) \end{aligned} \tag{6.32}$$

step 2. The control sequence $\overset{\star}{u}{}_k^{k+N_p-1}$ is then given by (from (6.32) for $\lambda = \lambda^\star$):

$$\overset{\star}{u}{}_k^{k+N_p-1} = [\overset{\star}{u}{}_{k-1}^\top, \ldots, \overset{\star}{u}{}_{k-1}^\top]^\top + T_{N_p} z(\lambda^\star) \tag{6.33}$$

The saturation is then applied a posteriori to the first control input before its application to the plant.

6.2.4 Numerical results

The following section illustrates the properties of the modified LRMPC formulation using the process described in section 4.4. To illustrate its benefits with respect to measurement noise rejection, a uniform random noise of magnitude 0.2 $g.L^{-1}$ at the maximum is added to the biomass measurements. The predictive tuning parameters are chosen as follows: N_p=6, T_s=5 min, $T_d = 0.1$ min, $V = \mathbb{I}_{N_p}$ and $W = 0.01\mathbb{I}_{N_p}$. In this example, more weight is given to V in order to promote the smoothness of the control input. The chosen values of V and W take into account the magnitude order of the control evolution in the cost function (smaller than the tracking error on the output).

First, simulations have been carried out considering the nominal case (no uncertainties) in order to focus only on the effect of the measurement noise. The achieved results are shown in Figures 6.1-6.3.
The obtained results in Figure 6.3 show that the control evolution is smoother when considering a penality on the control evolution (LRMPC vs LRMPC-δu). The tracking performances are quite similar.

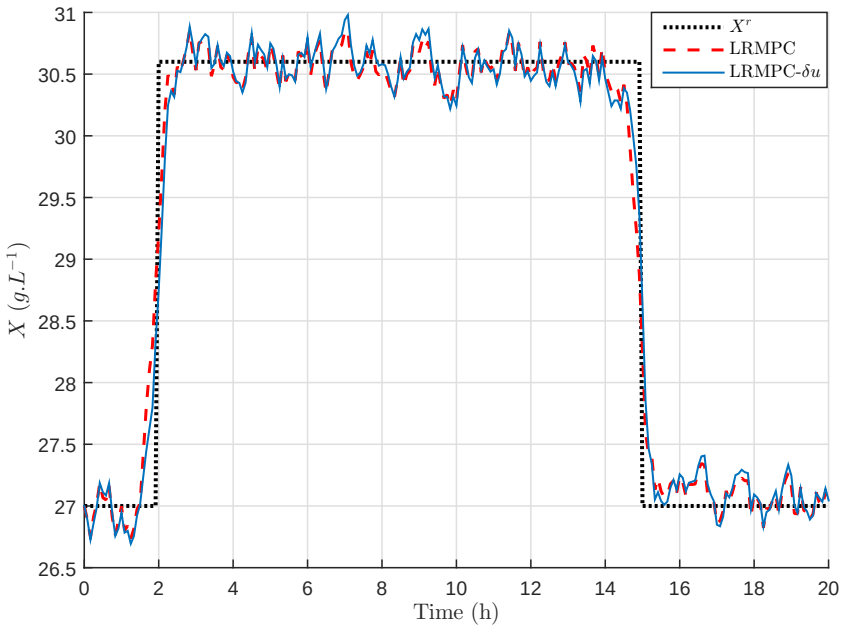

Figure 6.1: Biomass concentration evolution with time for LRMPC and LRMPC-δu strategies (nominal case).

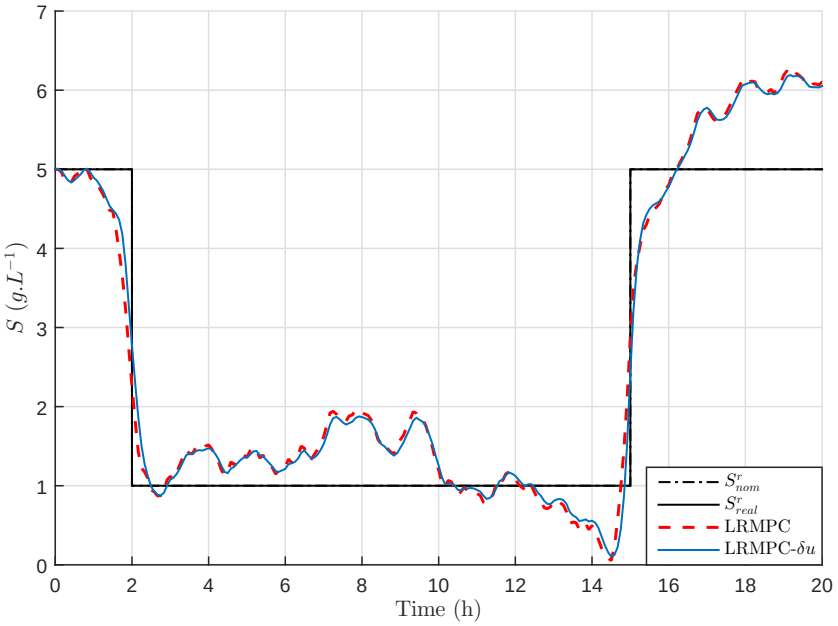

Figure 6.2: Substrate concentration evolution with time for LRMPC and LRMPC-δu strategies (nominal case).

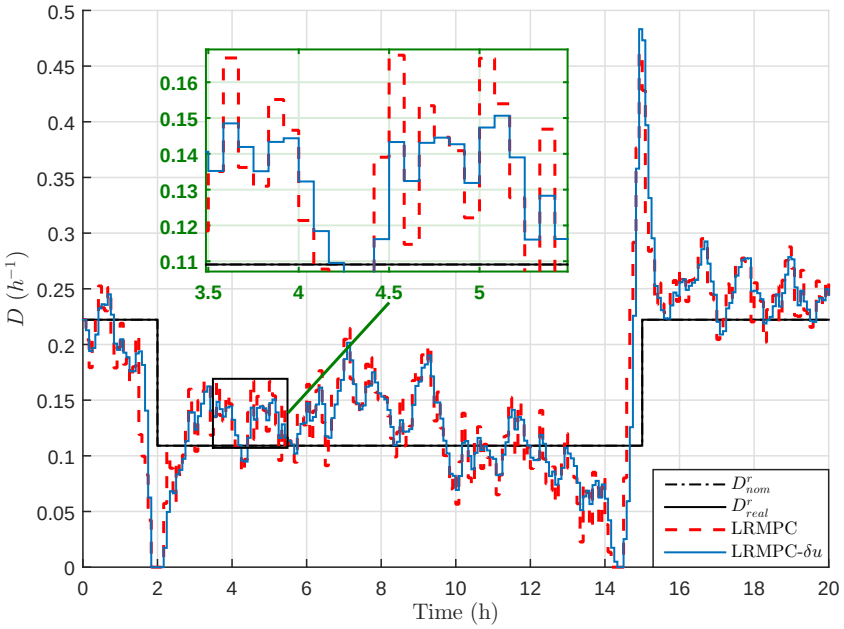

Figure 6.3: Control input evolution with time for LRMPC and LRMPC-δu strategies (nominal case).

Secondly, the achieved results shown in Figures 6.4-6.6 compare the performances of LRMPC and LRMPC-δu, with simulation conditions similar to those considered previously except we consider the uncertain parameter case discussed in the chapter 4 (see section 4.4.2).
It can be observed that the control evolution is smoother for LRMPC-δu in comparison with the LRMPC (as shown in Figure 6.6) but the tracking accuracy on the biomass concentration is decreased (Figure 6.4).

Indeed, there is a trade-off between tracking accuracy and control smoothness. This decrease of accuracy observed with LRMPC-δu can be overcome if another tuning of V and W is considered.

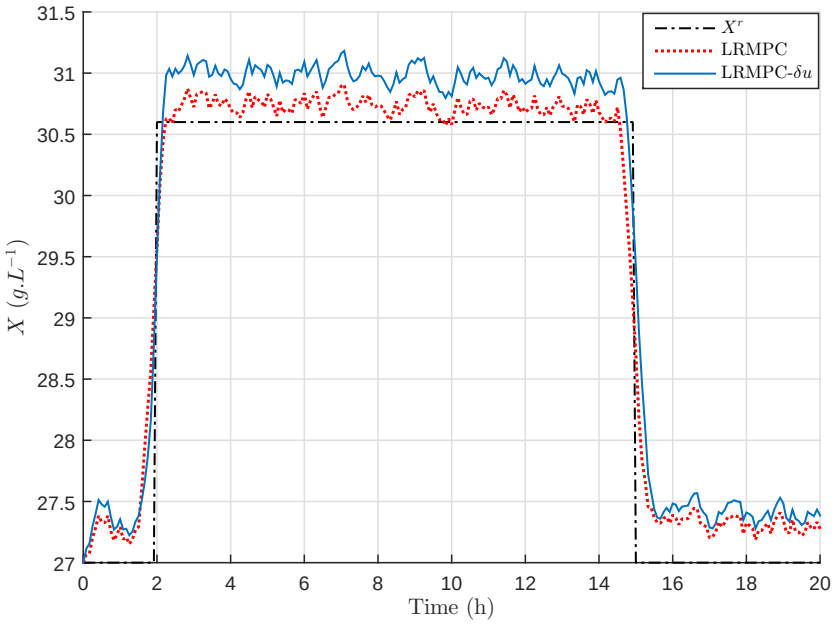

Figure 6.4: Biomass concentration evolution with time for LRMPC and LRMPC-δu strategies (mismatched parameters).

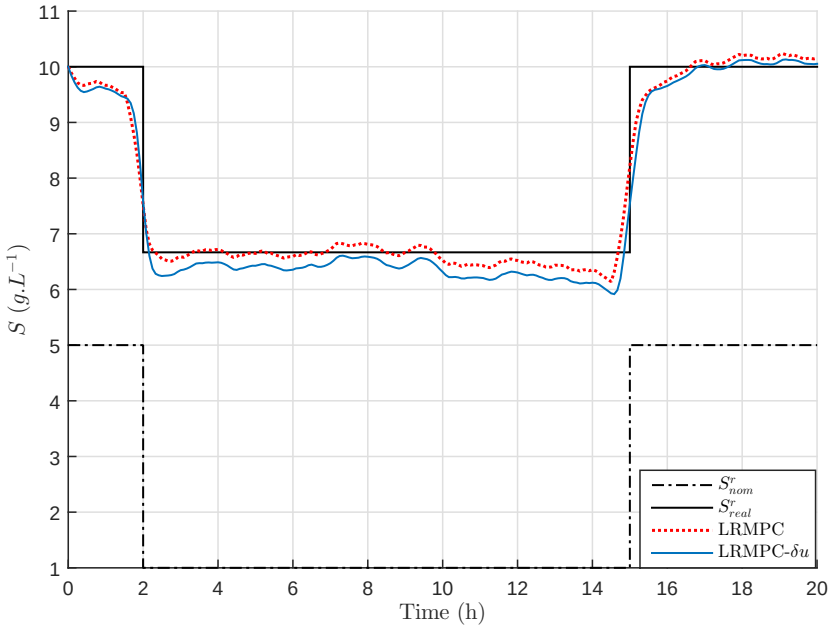

Figure 6.5: Substrate concentration evolution with time for LRMPC and LRMPC-δu strategies (mismatched parameters).

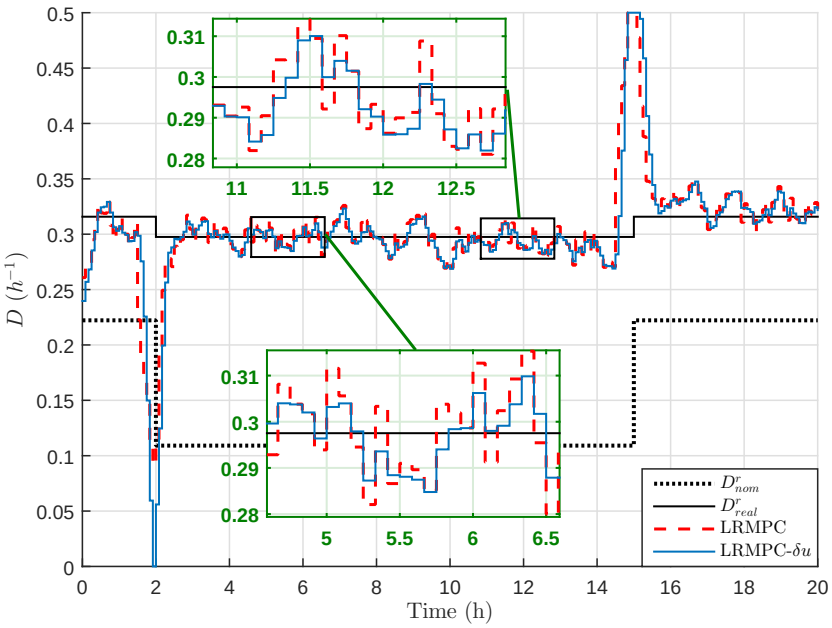

Figure 6.6: Control input evolution with time for LRMPC and LRMPC-δu strategies (mismatched parameters).

6.3 Hierarchical control strategy

As mentionned previously, the proposed robust predictive approaches can have accuracy issues especially for large sampling period (since they deal with a first-order approximation of nonlinear functions). In order to better handle this approximation, an interesting solution to increase the quality of the linearized model, may be considering a second order expansion rather than the first order approximation, as proposed in [61] for robust state estimation. Hereafter, we propose to use as an alternative strategy a hierarchical control scheme as depicted in Figure 6.7. This structure is similar to the Internal Model Control (IMC) (even if their principles are quite different) [59]. The controller is formed by a robust predictive law (control input denoted $u^\star$) coupled to an auxiliary controller (control input denoted $\tilde{u}$). The predictive controller allows tracking the reference trajectory, whereas the additional controller is added to cancel any residual tracking error, also taking into account that this additional controller should be as simple as possible in order not to increase the calculation time too much.

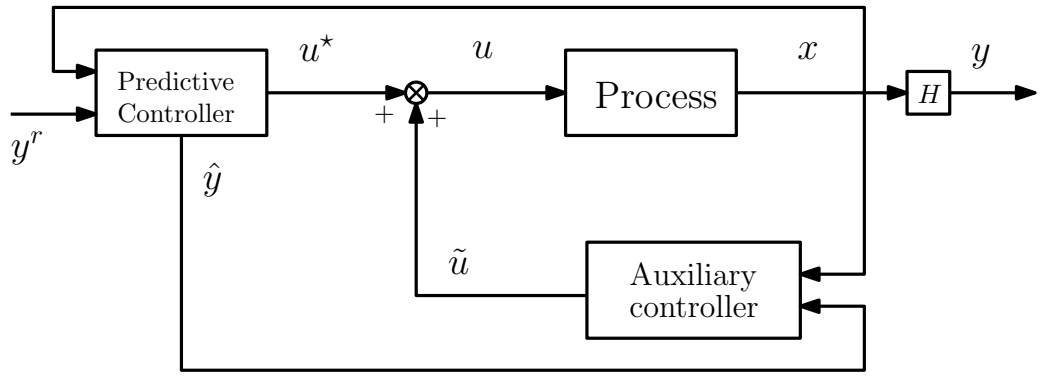

Figure 6.7: Scheme of the hierarchical control strategy [131].

In the sequel, two controllers will be investigated: the Proportional-Integral (PI) law and the Integral Sliding Mode (ISM) controller.

***Remark* 6.2.** The proposed hierarchical scheme can be used for all the predictive controllers (NMPC, RNMPC, rRNMPC, (C)LRMPC and LRMPC-δu) but in this section the hierarchical approach will be only presented in the case of LRMPC.

6.3.1 PI controller design

The proposed control strategy consists in a hierarchical control scheme (see Figure 6.8). The controller is formed by a robust predictive law (LRMPC)

coupled to a Proportional-Integral (PI) controller. The predictive controller allows tracking the reference trajectory, whereas the PI is added to reduce any residual tracking error. The motivation for choosing the PI controller is the simplicity of the control design in comparison with other advanced control strategies. In fact, the PI is a widely used control strategy that has its dynamics depending on the values of two constant gains: proportional and integral.

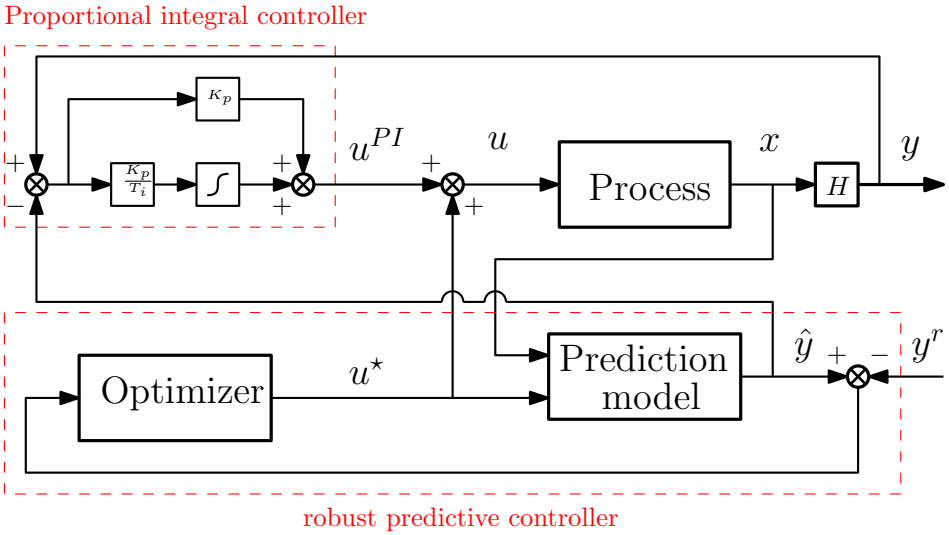

Figure 6.8: Scheme of the hierarchical control strategy based on robust MPC and PI controller.

At each time instant $t_k = kT_s$, the optimal control law $\overset{\star}{u}_k$ obtained from the predictive controller is completed by a Proportional-Integral law $\tilde{u} = u^{PI}$. The idea is to drive the system to track the predicted output $\hat{y}$ in order to cancel the difference between the model prediction output and the system output. The PI design is performed in continuous-time, then discretized for implementation. Let us define the model prediction resulting from the application of the previous control input for $t \in [t_{k-1}, t_k]$, as follows:

$$\hat{y}(t) = Hg(t_{k-1}, t, x_{k-1}, u_{k-1}, \theta_{\text{nom}}) \tag{6.34}$$

In the case of the considered control problem, the control law, $\tilde{u}$, derived from the PI strategy is given by (with K_p and T_i the controller parameters and are its tuning parameters):

$$\tilde{u}(t) = K_p(y(t) - \hat{y}(t)) + \frac{K_p}{T_i}\int_{t_{k-1}}^{t} (y(\tau) - \hat{y}(\tau))d\tau \tag{6.35}$$

Finally, with (6.35) evaluated at $t = t_k$ and a discretization via an Euler scheme, the control input to be applied to the plant is obtained as the sum of two parts, given by:

$$u(t_k) = \overset{\star}{u}_k + \tilde{u}(t_k) \tag{6.36}$$

The component $\overset{\star}{u}_k$ (the first value of the optimal control sequence) is generated by the LRMPC controller, while $\tilde{u}(t_k)$ is generated by the PI controller.

The main drawback of this controller is the tuning of the PI controller (especially to ensure the closed-loop plant stability).

6.3.2 ISM controller design

An alternative to the implementation of a PI controller is to use the hierarchical control scheme formed by a Sliding Mode Control (SMC) law and a robust predictive controller as shown in Figure 6.9 similarly to the one proposed in [131], which may in some cases improve performances compared to LRMPC-PI.

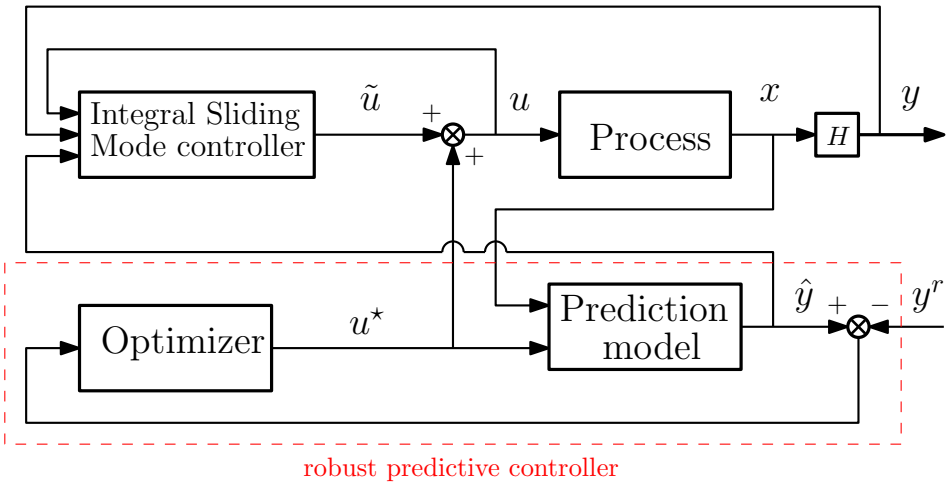

Figure 6.9: Scheme of the hierarchical control strategy based on robust MPC and ISM controller.

The reasons for the choice of the SMC are:

- it is a robust control technique which guarantees the complete elimination of the tracking error due to the approximation of the model through the linearization;
- it offers a strong disturbance rejection;
- it guarantees the finite-time convergence to the sliding manifold.

Details related to SMC can be found in [146]. The SMC considered in this study is designed according to the so-called Integral Sliding Mode (ISM) approach. In the latter case, an integral action is included in the SMC.

Let us assume that the considered system has a single output y and single input u and is control-affine, which is a special case of (4.1). We will then consider for the ISM development, the following model:

$$\begin{cases} \dot{x}(t) = f_x(x(t),\theta) + f_u(x(t),\theta)u(t), \quad \forall t > t_0, \quad x(t_0) = x_0 \\ y(t) = Hx(t) \end{cases} \tag{6.37}$$

At each time instant $t = kT_s$, the goal is to complete the optimal control law $\overset{\star}{u}$ obtained from the predictive controller by an Integral Sliding mode control law $\tilde{u}(t)$ in order to cancel the error between the predicted output and the system output (same philosophy as LRMPC-PI). In this study, the ISM design is performed in continuous-time, then discretized for implementation.

The sliding mode control design consists in choosing the control input in such a way that the system is driven to reach a sliding manifold and be maintained there for all future time. The goal is to track the predicted output $\hat{y}$ in order to cancel the difference between the model prediction output and the system output.

Let us define the modelling error variables for $t \in [t_{k-1}, t_k]$ [145]:

$$\begin{cases} Z_1(t) = \int_{t_{k-1}}^{t} (y(\tau) - \hat{y}(\tau))\,\mathrm{d}\tau \\ \\ Z_2(t) = \frac{1}{\xi_1}(y(t) - \hat{y}(t) - \xi_2 Z_1(t)) \end{cases} \tag{6.38}$$

where the model prediction $\hat{y}$ resulting from the application of the previous control input u_{k-1} is given by (6.34). $\xi_1 \in \mathbb{R}$ and $\xi_2 \in \mathbb{R}$ are the ISM tuning parameters.

Differentiating (6.38) with respect to time and rearanging (6.38), we obtain:

$$\begin{cases} \dot{Z}_1(t) = \xi_1 Z_2 + \xi_2 Z_1 \\ \\ \dot{Z}_2(t) = \frac{1}{\xi_1}(\dot{y}(t) - \dot{\hat{y}}(t) - \xi_2 \dot{Z}_1(t)) \end{cases} \tag{6.39}$$

A time-varying sliding surface $\phi(x,t)$ is defined in the state space $\mathbb{R}^{n_x}$ as

$$\phi(x,t) = Z_2(t) + \xi_3 Z_1(t) \tag{6.40}$$

The ISM control law needs to be designed so that the invariance of the sliding manifold is satisfied, *i.e.*:

$$\forall x \in \mathbb{R}^{n_x}, \phi(x,t) = 0 \tag{6.41}$$

Furthermore, the local attractivity of the sliding surface ϕ can be expressed by the condition:

$$\forall x \in \mathbb{R}^{n_x} \dot{\phi}(x,t)\phi(x,t) < 0 \tag{6.42}$$

Hence, the sliding surface (6.40) is made attractive by choosing [146]:

$$\dot{\phi}(x,t) = -K_s \text{sign}(\phi(x,t)) \tag{6.43}$$

where the switching gain K_s is a strictly positive constant.
Then, the attractive equation which implies that the distance to the sliding surface decreases along all system trajectories is satisfied since (from (6.42)-(6.43)):

$$\dot{\phi}(x,t)\phi(x,t) = -K_s|\phi(x,t)| < 0 \tag{6.44}$$

From (6.40) and (6.41), it comes:

$$Z_2(t) + \xi_3 Z_1(t) = 0 \tag{6.45}$$

Thanks to (6.39), equation (6.45) becomes:

$$\frac{1}{\xi_1}\dot{Z}_1(t) - \frac{\xi_2}{\xi_1}Z_1(t) + \xi_3 Z_1(t) = 0 \tag{6.46}$$

Then, we have that

$$\dot{Z}_1(t) = (\xi_2 - \xi_1\xi_3)Z_1(t) \tag{6.47}$$

In order to ensure the convergence of Z_1, the following condition must be satisfied:

$$\xi_2 - \xi_1\xi_3 < 0 \tag{6.48}$$

Consequently, differentiating the sliding surface vector (6.40), we obtain:

$$\dot{\phi}(x,t) = \frac{1}{\xi_1}(\dot{y}(t) - \dot{\hat{y}}(t) - (\xi_2 - \xi_1\xi_3)(\xi_1 Z_2(t) + \xi_2 Z_1(t))) \tag{6.49}$$

The system output is obtained by the application of the previous control input u_{k-1} (reminding that a ZOH is applied, *i.e.* $u(t) = u_{k-1}$, $\forall t \in [t_{k-1}, t_k]$) combined with the sliding mode control law $\tilde{u}(t)$ and the predicted output is calculated by applying only the previous input u_{k-1} as follows:

$$\begin{cases} \dot{y}(t) = H\left(f_x(x(t),\theta) + f_u(x(t),\theta)(u_{k-1} + \tilde{u}(t))\right) \\ \dot{\hat{y}}(t) = H\left(f_x(\hat{x}(t),\theta_{nom}) + f_u(\hat{x}(t),\theta_{nom})u_{k-1}\right) \end{cases} \tag{6.50}$$

where the difference between the system output and the nominal model prediction output is due only to parameters uncertainties:

$$\dot{y}(t) - \dot{\hat{y}}(t) = \varphi + \chi u_{k-1} + \eta\tilde{u}(t) \tag{6.51}$$

with

$$\begin{cases} \varphi = H(f_x(x(t),\theta) - f_x(\hat{x}(t),\theta_{nom})), \\ \chi = H(f_u(x(t),\theta) - f_u(\hat{x}(t),\theta_{nom})), \\ \eta = Hf_u(x(t),\theta) \end{cases} \tag{6.52}$$

From (6.51), it is deduced that the control law can be accordingly found as:

$$\tilde{u}(t) = \frac{1}{\eta}\left(\dot{y}(t) - \dot{\hat{y}}(t) - \varphi - \chi u_{k-1}\right) \tag{6.53}$$

Note that the term η must be regular. And from (6.49),

$$\dot{y}(t) - \dot{\hat{y}}(t) = \xi_1\dot{\phi} + (\xi_2 - \xi_1\xi_3)(\xi_1 Z_2(t) + \xi_2 Z_1(t)) \tag{6.54}$$

By replacing $\dot{\phi}$ by (6.43), the control law can then be found as

$$\tilde{u}(t) = \tfrac{1}{\eta}(-\varphi - \chi u_{k-1} - \xi_1 K_s \text{sign}(\phi(x,t)) + (\xi_2 - \xi_1\xi_3)(\xi_1 Z_2(t) + \xi_2 Z_1(t))) \tag{6.55}$$

In (6.52), θ is replaced by θ_{nom} since θ is uncertain. It could be replaced by an estimate of θ (with the use of an estimation algorithm). The use of θ in (6.55) is the main drawback of this approach. It should be recalled that the proposed controller can only be used to control-affine system which is another limitation of this strategy.

Another drawback of the ISM is the presence of a chattering phenomenon. In order to reduce it, a hyperbolic function can be used instead of the switching function $\text{sign}(\phi(x,t))$.

Finally, with (6.55) evaluated at $t = t_k$, the control input is obtained as the sum of two parts, given by:

$$u(t_k) = \overset{\star}{u}_k + \tilde{u}(t_k) \tag{6.56}$$

The component $\overset{\star}{u}_k$ (the first value of the optimal control sequence) is generated by the LRMPC controller, while $\tilde{u}(t_k)$ is generated by the Integral sliding mode controller (6.55).

6.3.3 Numerical results

In this section, the efficiency of the proposed hierarchical control strategies is evaluated in simulation. The performances of both the LRMPC-PI and LRMPC-ISM are compared for the worst uncertain parameters case discussed in the previous chapter (see section 4.4.2). The sampling time T_s is chosen equal to 5 min and T_d the integration time step to 0.1 min. The controllers tuning parameters are given in Table 6.2. They were obtained by a trial-and-error technique (the best setting found).

Table 6.2: Controllers tuning parameters.

	LRMPC			PI		ISM			
Parameter	N_p	V	W	K_p	T_i	ξ_1	ξ_2	ξ_3	K_s
Value	6	I_{N_p}	I_{N_p}	0.1	0.1	0.1	-0.1	2	15

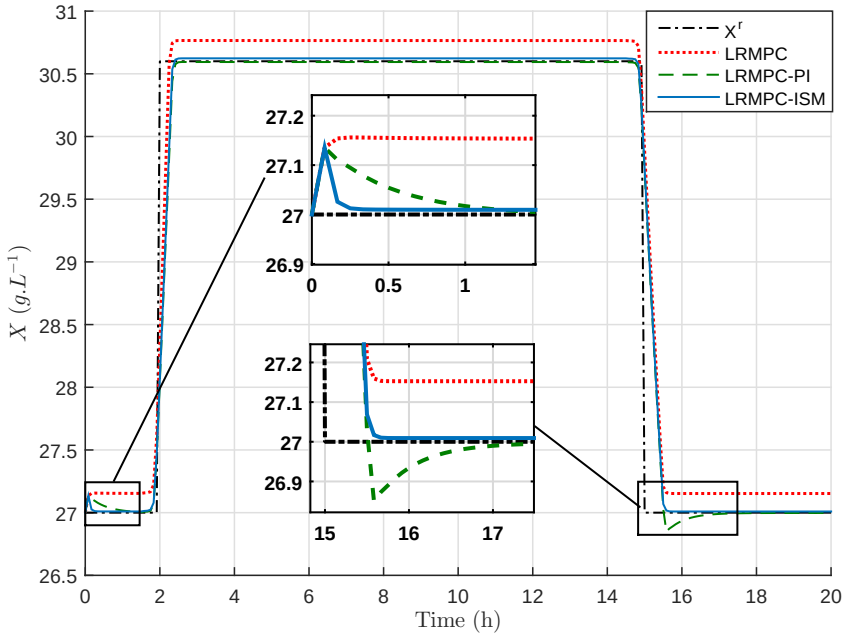

Figure 6.10: Biomass concentration evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

The obtained results as depicted in Figures 6.10-6.12 compare the LRMPC, the LRMPC-PI and the LRMPC-ISM controllers performances.

In the case of LRMPC law, it can be noticed the presence of a static error (Figure 6.10) due to the approximation of the model through linearization. On the other side, both the LRMPC-PI and LRMPC-ISM help to improve signifiantly the tracking accuracy in comparison with the LRMPC law. In fact, the dilution rate given by the hierarchical controllers moves away from its reference value D^r_{nom}, and converges to D^r_{real} as shown in Figure 6.12. In addition, it can be observed that the LRMPC-PI law leads to a larger overshoot in biomass concentration evolution during setpoint decrease in comparison with the LRMPC-ISM performances (Figure 6.10). The PI loop has a slower dynamic than the ISM one (Figure 6.12). Still, the overshoot is small and the LRMPC-PI presents a good global behavior.

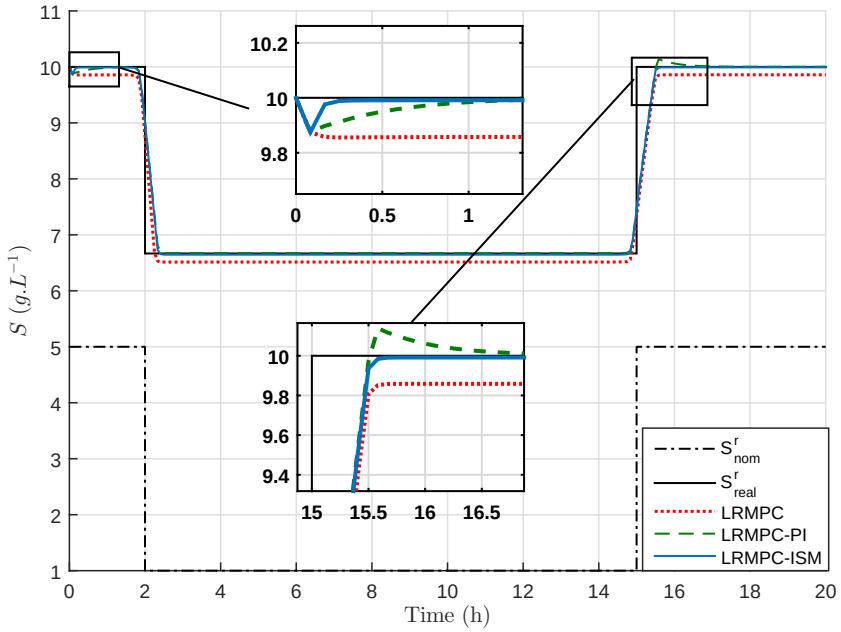

Figure 6.11: Substrate concentration evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

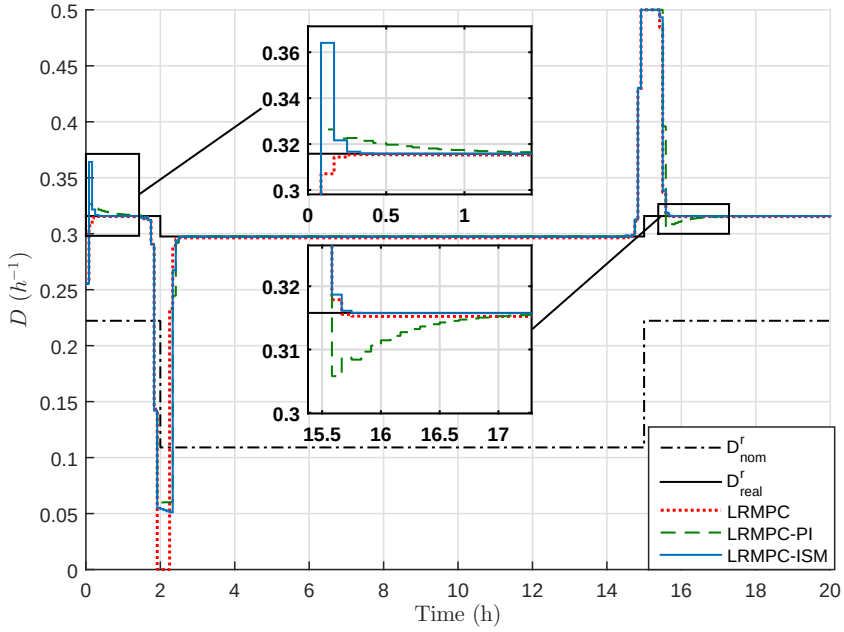

Figure 6.12: Control input evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

Thus, the hierarchical structure allows obtaining good closed-loop performances in terms of transient response and tracking accuracy.

6.4 Concluding remarks

In this chapter, two improvements have been proposed to the LRMPC controller. First, a new robust model predictive controller (denoted LRMPC-δu) is presented. The min max problem is turned into a more tractable optimization problem through the linearization of the predicted trajectory over the nominal parameters and the optimal contol sequence obtained at the previous sampling time. This leads to a control law less sensitive against noise measurements thanks to the inclusion of the penality term on the control evolution.

In a second step, a hierarchical control strategy is proposed. It consists in combining a robust predictive law (LRMPC) with an auxiliary controller (PI or ISM). The LRMPC allows tracking the reference trajectory, whereas the additional controller is added to cancel any residual tracking error. For the auxiliary controller, the choice of the PI presents the advantage of design simplicity in comparison with ISM controller which is a model-based control strategy. Nevertheless, the PI controller presents a slight loss of performances (response time) in comparison with the ISM regulator. To conclude, the PI law ensures the best trade-off between complexity of implementation and performances.

All control strategies presented till now will be tested in the case of a complex system (in comparison with the previous illustrative example). In the next chapter, the predictive controllers will be applied to control microalgae culture in a continuous photobioreactor.

Chapter 7

Illustrative example: Microalgae cultivation system

7.1 Introduction

One of the most important challenges of this century is to satisfy our energetic needs while addressing environemental issues. In the mean time, the influence of climate change, and the decline of fossil fuels have motivated a growing interest in the search of renewable sources of energy. The biotechnology industry is an important sector due to its economical, societal and environmental potential. Microalgae culture are among the most interesting types of biocultures. A microalga is a microscopic plant living mainly in aquatic environment, representing the basic level of the food chain in the ocean. These organisms have an increasing interest due to their ability to fix CO_2 [111] and to produce lipid or hydrogen that can be used as biofuel [33, 71, 101]. The industrial success of the microalgae cultivation is due to its biochemical characteristics. The microalga is of particular interest for the growing demand of organic products intended to a large number of industrial applications: feed, food, pharmacology, chemistry, cosmetics production [55, 140], and has recently emerged as an interesting source for sustainable energy production at large scale, such as wastewater treatment [56] or decomposition of different classes of toxic compounds [22]. For these reasons, microalgae cultivation attracted the interest from large companies. In this context, the control strategy of microalgae culture is becoming a key research topic and consequently it received high attention from scientific community leading to many studies. This requires first the design of mathematical models to describe more accurately the system behavior. It is however quite delicate to derive a simple model which can be exploited during online optimization

phases because the microalgae are living organisms and their characteristics can evolve in time (mutation, biofilm formation, stress, *etc.*). Biochemical processes are systems where nonlinear effects are significant enough to justify the use of nonlinear model to give a sufficiently adequate representation of the system behavior. In addition, generally, the process model is identified and uncertain parameters are estimated with evaluated confidence intervals, which motivates the development of robust control laws in the presence of modelling uncertainties.

In industrial applications, the control of bioreactors is most often limited to simple regulation loops like: partial pressure of dissolved oxygen and dissolved carbon dioxide [48], pH and temperature. Hence, to increase significantly the process performances, new challenges emerge related to the control of the biological variables. In the literature of microalgae cultivations, several nonlinear control strategies have been developed: predictive [1, 10, 143, 134, 47], adaptive [97, 38, 96], sliding mode [138], feedback linearization [11, 72, 142], and backstepping [53, 145] approaches. They however do not specifically focus on robustness features. Our aim is therefore to design a robust predictive controller which would be able to find an optimal feeding strategy in order to guarantee that the process will yield the desired amount of biomass along the cultivation period under model parameter uncertainties. Here, the challenge is to lay down a stable real-time operation, insensitive to various disturbances, then, close to a certain state or desired profile. This requires the application of advanced optimal control strategies to ensure the bioprocess efficiency.

In this chapter, the proposed approaches developped in the previous chapters are applied to control microalgae culture in photobioreactor through a specific case study: the cultivation of *I. galbana* in a continuous photobioreactor [98]. A Droop model is used to describe the internal nutrient quantity per unit of biomass evolution. The aim here is to control biomass concentration in the photobioreactor through the dilution rate.

7.2 System modelling

7.2.1 Model Description

The specificity of microalgae is that inorganic substrate uptake and growth are decoupled thanks to an intracellular storage of nutrients [15]. In order to take into account this phenomenon, the growth of microalgae is represented by a Droop model [16, 15] which uncouples growth from substrate uptake, leading to the definition of an internal cell quota (i.e., the internal nutrient

quantity per unit of biomass), and describes the growth rate as a function of the internal quota only. The mass balance model (see Appendix B) involves three state variables: the biomass concentration (denoted X, in $\mu m^3\ L^{-1}$), the internal quota (denoted Q, in $\mu mol\ \mu m^{-3}$), and the substrate concentration (denoted S, in $\mu mol\ L^{-1}$). The considered dynamic model assumes that the photobioreactor is in continuous mode (medium withdrawal flow rate equals its supply one, leading to a constant effective volume), without any additional biomass in the feed and neglecting the effect of gas exchanges. The time varying equations resulting from mass balances are given by [98]:

$$\begin{cases} \dot{X}(t) = \mu(Q(t), I(t))\, X(t) - DX(t) \\ \dot{Q}(t) = \rho(S(t)) - \mu(Q(t), I(t))\, Q(t) \\ \dot{S}(t) = (S_{in} - S(t))D - \rho(S(t))X(t) \end{cases} \tag{7.1}$$

where

- D represents the dilution rate (d^{-1}, d: day) which is the ratio of the inlet flow rate to the volume of the culture.
- S_{in} the input substrate concentration ($\mu mol\ L^{-1}$).

The specific uptake rate $\rho(S)$ is given by a Monod kinetic:

$$\rho(S) = \rho_m \frac{S}{S + K_s} \tag{7.2}$$

The parameters K_s and ρ_m represent respectively the substrate half saturation constant and the maximal specific uptake rate.
The specific growth rate $\mu(Q, I)$, on the other hand, can be defined as a function of the intercellular quota Q as follows (Droop model):

$$\mu(Q, I) = \bar{\mu} \left(1 - \frac{K_Q}{Q}\right) \mu_I(I) \tag{7.3}$$

The theoretical specific maximal growth rate (at infinite internal quota) is denoted $\bar{\mu}$ and K_Q represents the minimal cell quota, for which no algal growth can take place.

The light intensity has a direct effect on growth (photosynthesis), while uptake can continue in the dark. The modelling of light effect consists in including the term μ_I in (7.3) which is represented by a Haldane type kinetics to model the photoinhibition [114]:

$$\mu_I(I) = \frac{I}{I + K_{sI} + \frac{I^2}{K_{iI}}} \tag{7.4}$$

where I is the light intensity (μE m^{-2} s^{-1}) and K_{sI} and K_{iI} are light saturation and inhibition constants respectively. The optimal light intensity that maximises the function μ_I is given by $I_{opt} = \sqrt{K_{sI}K_{iI}}$. In the sequel, the light intensity is either set at this optimal value I_{opt} or is time varying.
The parameters of the model used in this study are given in Table 7.1 [61],[107].

Table 7.1: Droop model parameters.

Parameter	Value	Unit
$\bar{\mu}$	2	d^{-1}
ρ_m	9.3	μmol μm^{-3} d^{-1}
K_Q	1.8	μmol μm^{-3}
K_s	0.105	μmol L^{-1}
K_{sI}	150	μE m^{-2} s^{-1}
K_{iI}	2000	μE m^{-2} s^{-1}

7.2.2 Model analysis

The nonlinear model (7.1) is represented thereafter in the state-space formalism (4.1) as follows:

$$\begin{cases} \dot{x}(t) = F(x, u, w, \theta), \quad x(t_0) = x_0 \\ y = X \end{cases} \tag{7.5}$$

with

$$\begin{cases} x = \begin{bmatrix} X \\ Q \\ S \end{bmatrix}, \ w = \begin{bmatrix} S_{in} \\ I \end{bmatrix}, \ u = D \\ F = \begin{bmatrix} \mu(Q, I)X - DX \\ \rho(S) - \mu(Q, I)Q \\ (S_{in} - S)D - \rho(S)X \end{bmatrix} \\ \theta = \begin{bmatrix} \rho_m & K_s & \bar{\mu} & K_Q & K_{sI} & K_{iI} \end{bmatrix}^T \end{cases} \tag{7.6}$$

where the state variables are assembled in a vector denoted x and x_0 its initial value. The nonlinear process dynamics is denoted F. The measurements are related to the vector y whereas the inputs are represented by the vector u. The other exogenous inputs are denoted w. Finally, the parameters refer to the vector θ.
The maximum growth rate denoted μ_m is reached for the maximum internal quota Q_l. From (7.3), it comes:

$$\mu_m = \mu(Q_l, I) = \bar{\mu}\left(1 - \frac{K_Q}{Q_l}\right)\mu_I \tag{7.7}$$

From the dynamics of Q, the maximal specific uptake is related to μ_m by:

$$\rho_m = \mu_m Q_l \tag{7.8}$$

From (7.7)-(7.8), the maximum cell quota Q_l obtained in conditions of non-limiting nutrient is given by:

$$Q_l = \frac{\rho_m}{\bar{\mu}\mu_I} + K_Q \quad \text{with} \quad \bar{\mu}\mu_I \neq 0 \tag{7.9}$$

The condition $\bar{\mu}\mu_I \neq 0$ holds $\forall I \neq 0$ which is the case here.
In addition, the cell quota is constrained to remain between two bounds [15]:

$$K_Q \leq Q \leq Q_l \tag{7.10}$$

The growth rate is also bounded:

$$0 \leq \mu(Q) \leq \mu_m \tag{7.11}$$

where (by replacing (7.9) in (7.8))

$$\mu_m = \frac{\rho_m \bar{\mu}\mu_I}{\rho_m + K_Q \bar{\mu}\mu_I} \tag{7.12}$$

Finally, the state and control variables are restricted to fulfill the following constraints [16]:

$$\begin{cases} X > 0 \\ K_Q \leq Q \leq \dfrac{\rho_m}{\mu_I \bar{\mu}} + K_Q \\ 0 \leq S < S_{in} \\ D \geq 0 \end{cases} \tag{7.13}$$

7.2.3 Determination of equilibrium

The steady states of the system are derived from three nonlinear equations, given in order to cancel out the model's dynamic equations, *i.e.*

$$F(x_e, u_e, w, \theta) = 0 \tag{7.14}$$

where w is assumed constant and known). For a given value of X, the goal here is to characterize the corresponding values for Q, S and D for a constant light intensity I. Then, the equilibrium points are defined as follows (from (7.6)):

$$\begin{cases} \mu(Q_e, I) - D_e = 0 \\ \rho(S_e) - \mu(Q_e, I) Q_e = 0 \\ (S_{in} - S_e) D_e - \rho(S_e) X_e = 0 \end{cases} \tag{7.15}$$

Then, rearranging (7.15), the following system of equations has to be solved algebraically :

$$\begin{cases} D_e = \bar{\mu}\left(1 - \dfrac{K_Q}{Q_e}\right)\mu_I \\ \\ S_e = \dfrac{\mu(Q_e, I)Q_e K_s}{\rho_m - \mu(Q_e, I)Q_e} \\ \\ \left(S_{in} - \dfrac{\mu(Q_e, I)Q_e K_s}{\rho_m - \mu(Q_e, I)Q_e}\right) - Q_e X_e = 0 \end{cases} \tag{7.16}$$

Taking Q_e as an unknown variable, the following quadratic equation must be solved:

$$\begin{aligned} &[\bar{\mu}\mu_I X_e]Q_e^2 - [(S_{in} + K_s)\bar{\mu}\mu_I + (\rho_m + \bar{\mu}\mu_I K_Q)X_e]Q_e \\ &+ [(\rho_m + \bar{\mu}\mu_I K_Q)S_{in} + \bar{\mu}\mu_I K_Q K_s] = 0 \end{aligned} \tag{7.17}$$

The discriminant of equation (7.17) is given by:

$$\Delta = aX_e^2 + bX_e + c = (\sqrt{a}X_e - \sqrt{c})^2 + (2\sqrt{a}\sqrt{c} + b)X_e \tag{7.18}$$

where

$$\begin{cases} a = (\rho_m + \bar{\mu}\mu_I K_Q)^2 > 0 \\ b = 2\bar{\mu}\mu_I\left((K_s - S_{in})(\rho_m + \bar{\mu}\mu_I K_Q) - 2\bar{\mu}\mu_I K_Q K_s\right) \\ c = \bar{\mu}^2\mu_I^2(S_{in} + K_s)^2 > 0 \end{cases} \tag{7.19}$$

We have that

$$\begin{aligned} 2\sqrt{a}\sqrt{c} + b =& 2(\rho_m + \bar{\mu}\mu_I K_Q)\bar{\mu}\mu_I(S_{in} + K_s) \\ &+ 2\bar{\mu}\mu_I\left((K_s - S_{in})(\rho_m + \bar{\mu}\mu_I K_Q) - 2\bar{\mu}\mu_I K_Q K_s\right) \\ 2\sqrt{a}\sqrt{c} + b =& 4\bar{\mu}\mu_I K_s \rho_m > 0 \end{aligned} \tag{7.20}$$

Finally, it holds

$$\Delta = \left((\rho_m + \bar{\mu}\mu_I K_Q)X_e - \bar{\mu}\mu_I(S_{in} + K_s)\right)^2 + 4\bar{\mu}\mu_I K_s \rho_m X_e \tag{7.21}$$

Since the discriminant Δ is strictly positive (and reminding that $X_e > 0$, see (7.13)), there are two real solutions for (7.17):

$$Q_{1,2}^* = \frac{S_{in} + K_s}{2X_e} + \frac{1}{2}\left(\frac{\rho_m}{\bar{\mu}\mu_I} + K_Q\right) \mp \frac{\sqrt{\Delta}}{2\bar{\mu}\mu_I X_e} \tag{7.22}$$

By using equations (7.9), (7.18) and (7.19), we obtain:

$$\frac{\sqrt{\Delta}}{2\bar{\mu}\mu_I X_e} = \frac{1}{2}Q_l\sqrt{1+\alpha\frac{1}{X_e}+\beta\frac{1}{X_e^2}} \tag{7.23}$$

with

$$\begin{cases} \alpha = \dfrac{b}{a} \\ \\ \beta = \dfrac{c}{a} > 0 \end{cases} \tag{7.24}$$

Then (7.22) becomes:

$$Q^*_{1,2} = \frac{S_{in}+K_s}{2X_e} + \frac{1}{2}Q_l\left(1 \mp \sqrt{1+\alpha\frac{1}{X_e}+\beta\frac{1}{X_e^2}}\right) \tag{7.25}$$

From (7.13) the choice of the equilibrium X_e must be chosen to fulfill the following conditions:

- **Condition $Q^*_{1,2} \geq 0$:**

$$\begin{cases} Q^*_1 \geq 0 \text{ iff } \left(X_e - \sqrt{X_e^2+\alpha X_e+\beta}\right) \geq -\dfrac{S_{in}+K_s}{Q_l} & (7.26) \\ \\ Q^*_2 \geq 0,\ \forall X_e & (7.27) \end{cases}$$

- **Condition $Q^*_{1,2} \leq Q_l$:**

$$\begin{cases} Q^*_1 \text{ iff } \left(X_e + \sqrt{X_e^2+\alpha X_e+\beta}\right) \geq \dfrac{S_{in}+K_s}{Q_l} & (7.28) \\ \\ Q^*_2 \text{ iff } \left(X_e - \sqrt{X_e^2+\alpha X_e+\beta}\right) \geq \dfrac{S_{in}+K_s}{Q_l} & (7.29) \end{cases}$$

- **Condition $0 \leq S_e < S_{in}$:**

$$\rho_m \geq \mu(Q_e, I)Q_e \tag{7.30}$$

  which is equivalent to have $Q_e \leq Q_l$ (and hence $Q^*_{1,2} \leq Q_l$ which is the previous condition).

7.3 Control strategy

The main objective of the controller is to regulate the biomass concentration X to a reference value X^r in the presence of parameters uncertainties and noise measurement, while the dilution rate D is constrained to track the reference D^r where $0 \leq D \leq D_{\max}$ ($D_{\max}$ is the maximal dilution rate).

The controllability of this model was checked (See Appendix C).
The NMPC problem is then formulated as:

$$\min_{0 \leq D_k^{k+N_p-1} \leq D_{\max}} ||D_k^{k+N_p-1} - D_k^{r,k+N_p-1}||_V^2 + ||X_{k+1}^{k+N_p} - X_{k+1}^{r,k+N_p}||_W^2 \quad (7.31)$$

and the RNMPC by:

$$\min_{0 \leq D_k^{k+N_p-1} \leq D_{\max}} \max_{\theta \in [\theta^-, \theta^+]} ||D_k^{k+N_p-1} - D_k^{r,k+N_p-1}||_V^2 + ||X_{k+1}^{k+N_p} - X_{k+1}^{r,k+N_p}||_W^2 \quad (7.32)$$

where the uncertain parameters subspace $[\theta^-, \theta^+]$ is given by $[0.7\theta_{nom}, 1.3\theta_{nom}]$ with θ_{nom} given in Table 7.1. This 30% mismatch has been chosen as a rather classical percenttage. A more rigourous approach could be to proceed with an identification procedure as in [24] to determine the confidence intervals for all parameters.

The performance of the controllers in case of disturbances (here the light intensity fluctations) will be also studied and evaluated.

7.4 Simulation results

The efficiency of the proposed control strategies is validated in simulation. The initial biomass concentration value is set close to the setpoint in order to cancel the transient effect and focus only on the behavior during setpoint changes (rising and falling edge respectively). The light intensity is assumed to be non-measured, non-corrupted with noise. In sections 7.4.2 and 7.4.3, it is constant, equal to I_{opt} that maximizes $\mu(Q, I)$ defined in (7.3). In section 7.4.4, it is time-varying. The performances of the predictive algorithms presented in chapters 4, 5 and 6 are compared in a worst uncertain parameters case. Thanks to the monotonocity properties of the Droop model as discussed in [61], the worst-case prediction can be generated using parameters bounds $\{\theta^-, \theta^+\}$ only, rather than by exploring the full parameter space $[\theta^-, \theta^+]$. The parameters values of the system are chosen on the parameter subspace border ($\theta_{real} = [\rho_m^+, K_s^-, \bar{\mu}^+, K_Q^-, K_{sI}^-, K_{iI}^+]$). All the simulation conditions are summarized in Table 7.2.

Table 7.2: Simulation conditions for the Droop model.

	Variable	Value	Unit
sampling time	T_s	10	min
integration time step	T_d	12	sec
simulation time	T_f	1	d
inlet substrate concentration	S_{in}	100	μmol L^{-1}
optimal light intensity	I_{opt}	547	$\mu\mathrm{E}\ \mathrm{m}^{-2}\ \mathrm{s}^{-1}$
maximum cell quota	Q_l	9	μmol $\mu\mathrm{m}^{-3}$
maximal admissible dilution rate	$D_{\max}$	1.6	d^{-1}
prediction horizon	N_p	5	-
weighting matrix on control	V	$\mathbb{I}_{N_p}$	-
weighting matrix on state	W	$\mathbb{I}_{N_p}$	-
initial biomass concentration	$X(0)$	24.95	$\mu\mathrm{m}^3\ \mathrm{L}^{-1}$
initial internal quota	$Q(0)$	4	μmol $\mu\mathrm{m}^{-3}$
initial substrate concentration	$S(0)$	0.05	μmol L^{-1}

7.4.1 Determination of the reference

Two configurations may be considered for the determination of the pair (D^r, X^r) of reference signals:

- **Case 1:** $D^r \longrightarrow X^r$
  The biomass reference trajectory X^r is obtained by applying the dilution rate reference trajectory D^r to the model.

- **Case 2:** $X^r \longrightarrow D^r$
  In case of constant X^r (with an assumed constant light intensity), the dilution rate reference trajectory D^r is computed from the knowledge of the target setpoint X^r using relations at equilibrium (7.16):

$$D^r = \bar{\mu}\left(1 - \frac{K_Q}{Q^r}\right)\mu_I \tag{7.33}$$

  with Q^r the reference internal quota given by (7.25). From (7.25), two admissible solutions are possible for a given X^r, depending on the chosen setpoint.

  In case of time-varying X^r or X^r constant with a time-varying light intensity, the dilution rate reference trajectory D^r could be determined by solving the following constrained open-loop optimization problem:

$$D^r(t) = \arg \min_{0 \le D(t) \le D_{\max}} |X(t) - X^r(t)|^2 \tag{7.34}$$

In the following simulations, only case 2 will be considered, starting with the knowledge of the reference biomass concentration X^r.

Figures 7.1, 7.2 and 7.3 show the graphical representation of the conditions (7.26), (7.28) and (7.29) for the considered numerical data (Tables 7.2 and 7.1).

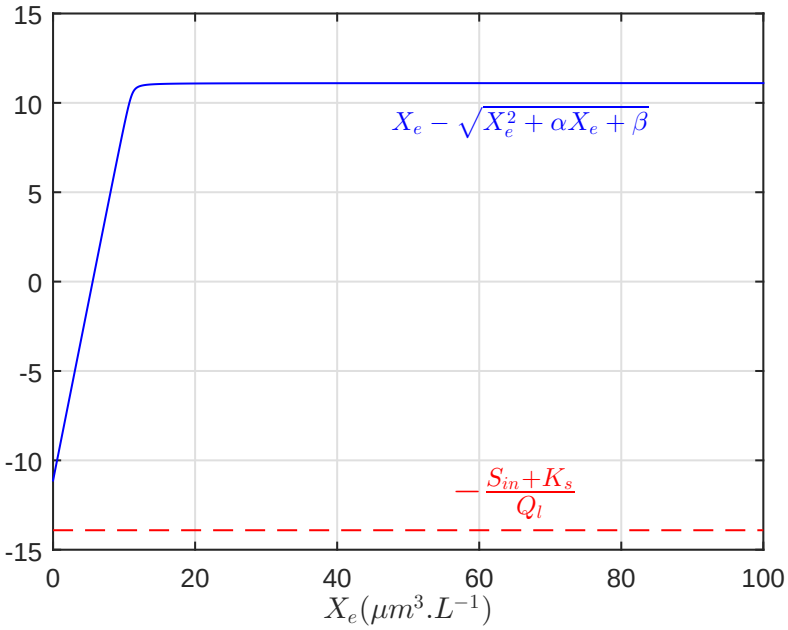

Figure 7.1: Graphical representation of the condition (7.26).

In one side, it can be observed that Q_1^* is always non-negative (Figure 7.1) and lower than Q_l (condition 7.2 holds) for values of $X_e > X_l \approx 12.5\ \mu\text{m}^3\ \text{L}^{-1}$ as depicted in Figure 7.2 (green box). On the other side, it appears clearly that Q_2^* is a non-admissible solution since it does not respects the condition (7.29) as shown in Figure 7.3.

Then, to have a feasible solution (here only one, which is Q_1^*), the reference biomass concentration X^r should satisfy the following constraint:

$$X^r \geq X_l \tag{7.35}$$

In our case, if the reference is constant, X^r must be chosen greater than $X_l \approx 12.5\ \mu\text{m}^3\ \text{L}^{-1}$. In the following, during constant tracking of a setpoint, X^r will be chosen between 25 and 27 $\mu\text{m}^3\ \text{L}^{-1}$.

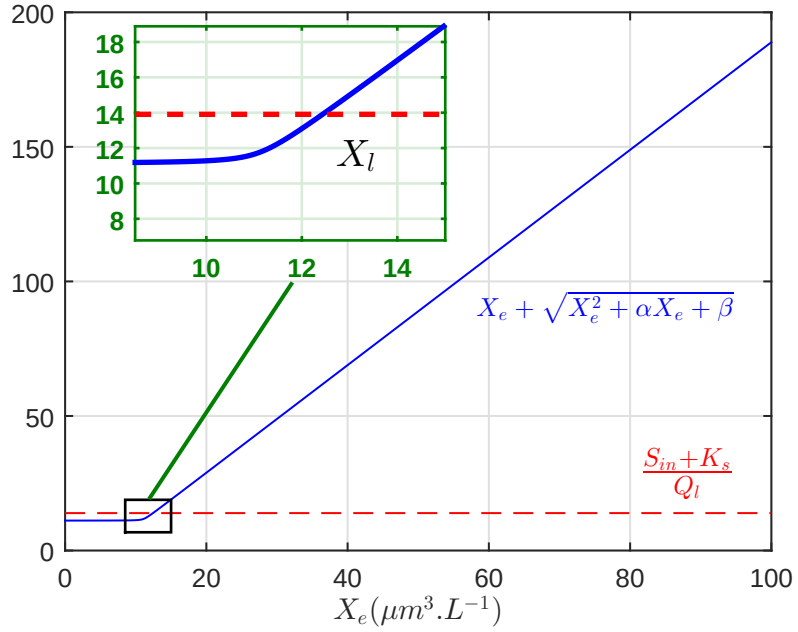

Figure 7.2: Graphical representation of the condition (7.28).

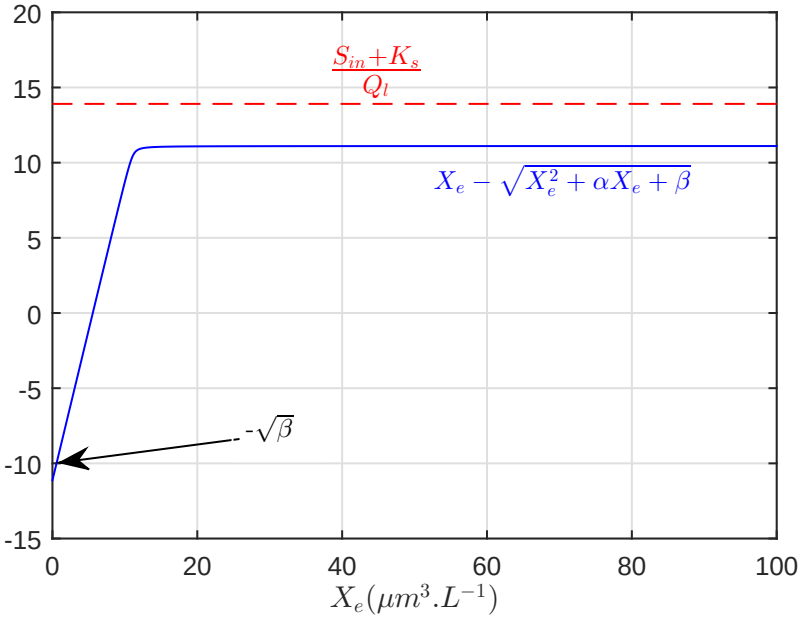

Figure 7.3: Graphical representation of the condition (7.29).

7.4.2 Setpoint tracking

First, the performances of setpoint tracking is studied. The goal is to track biomass concentration setpoint (rising and falling step changes as illustrated in Figure 7.4). Four predictive control laws are tested (Figure 7.5): a classical Nonlinear Model Predictive Control (4.12) (denoted as NMPC), a robust NMPC using criterion (5.3) (denoted as RNMPC), a reduced robust one using criterion (5.11) (denoted as rRNMPC) and the linearized one (5.104)-(5.105) (LRMPC). Biomass concentration measurements, y_k, are assumed to be corrupted by a centred Gaussian white noise with 0.1 standard deviation. Thanks to the sensitivity analysis of the model with respect to its parameters (see Appendix C), only the most influential parameters κ are considered in the min-max problem instead of the full model parameters as presented in section 5.3 (rRNMPC). In this case, the substrate half saturation constant K_s and the minimal cell quota K_Q are the most influential parameters on the biomass concentration evolution (*i.e.* $\kappa = [K_s, K_Q]^\top$). The other parameters, $\zeta = [\rho_m, \bar{\mu}, K_{sI}, K_{iI}]^\top$, are set to their nominal values (Table 7.1).

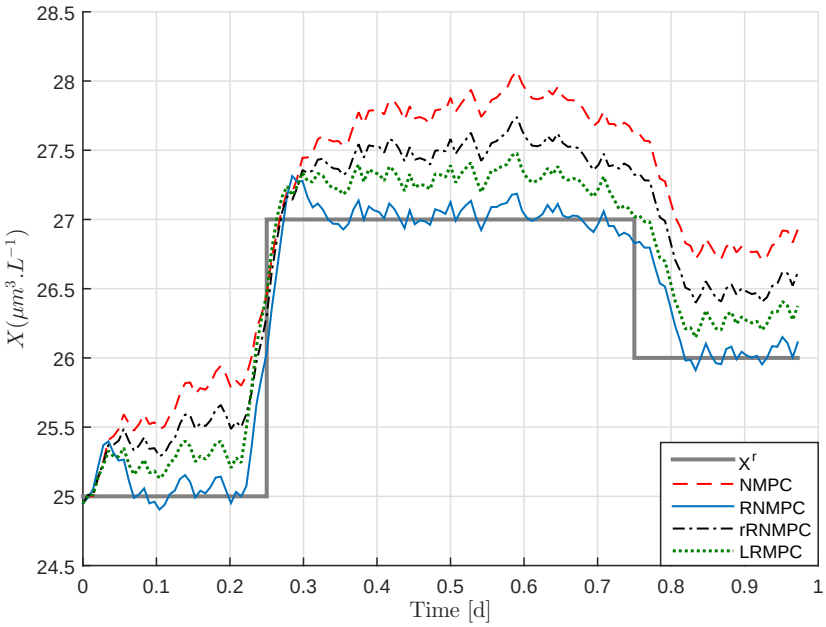

Figure 7.4: Biomass concentration evolution with time for NMPC, RNMPC, rRNMPC and LRMPC strategies.

It can be noticed the anticipation behavior to a setpoint change (Figure 7.4) for all controllers, due to the prediction of the setpoint trajectory future evolution over the moving horizon.

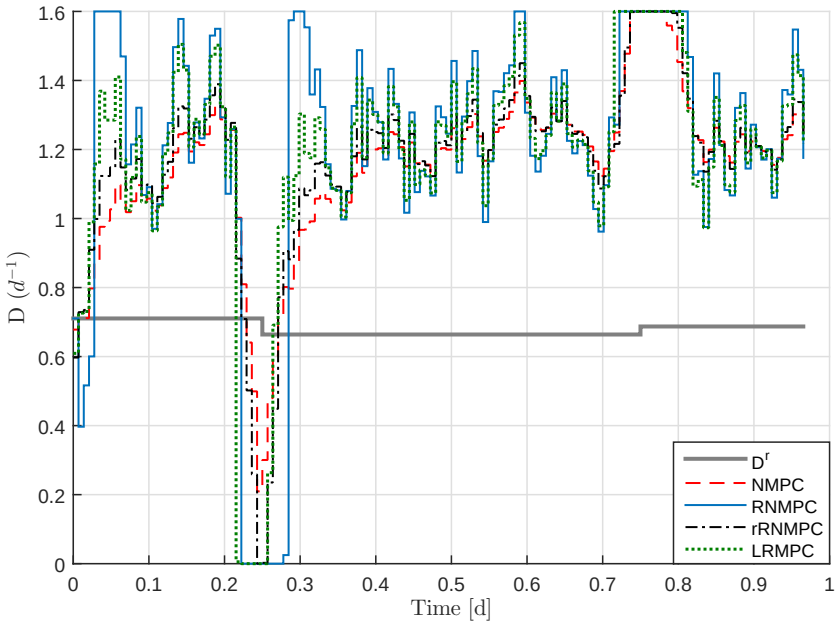

Figure 7.5: Dilution rate evolution with time for NMPC, RNMPC, rRNMPC and LRMPC strategies.

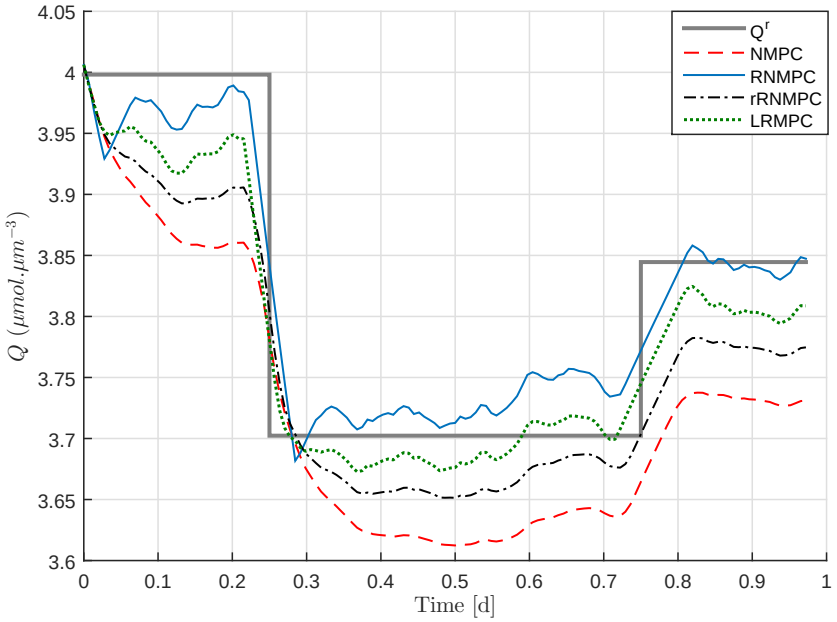

Figure 7.6: Internal quota evolution with time for NMPC, RNMPC, rRNMPC and LRMPC strategies.

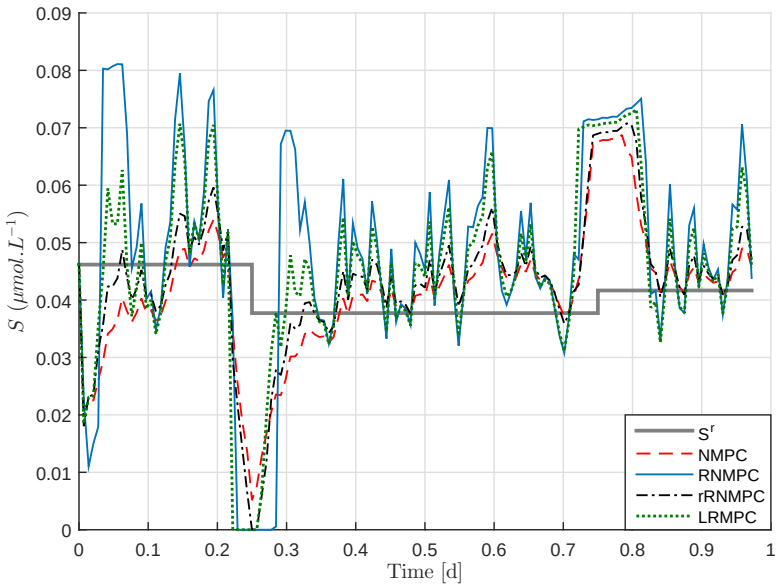

Figure 7.7: Substrate concentration evolution with time for NMPC, RNMPC, rRNMPC and LRMPC strategies.

The dilution rate decrease leads to an increase of the cell concentration (*e.g.* between 0.2 d and 0.3 d in Figure 7.5) and vice versa (*e.g.* between 0.7 d and 0.8 d), which agrees with the biological aspect. The obtained results show that both (r)RNMPC and LRMPC have better performances than the classical NMPC under parameter uncertainties and measurement noise. In the NMPC law, the biomass concentration is not able to track the specified setpoint in the presence of parameters uncertainties, due to the fact that the mismatch between the system and the model is not considered during the prediction step inside the minimization procedure. Furthermore, the RNMPC has better performances than the rRNMPC and LRMPC controllers under parameter uncertainties in term of tracking accuracy. The LRMPC performs better than the rRNMPC since it considers all model parameters. Furthermore, the classical RNMPC has better results than the reduced one but the computational burden of the former is much higher. The LRMPC algorithm performs well and offers a very significant computational load reduction comparing with (r)RNMPC as shown in Table 7.3.

***Remark* 7.1.** A comparison between NMPC-$\varepsilon^{s/m}$ and Generic Model Control (GMC [82]) is provided in Appendix C. The latter is studied since it is widely used for bioprocess control. The predictive controller presents better performances than the GMC law, highlighting the benefits of the proposed controller.

Table 7.3: Comparison of the predictive algorithms in terms of computation time at each sampling time.

Algo. \ Perf. indices	Computation time (s)		
	min	mean	max
NMPC	$< 10^{-5}$	0.014	0.29
RNMPC	0.55	1.85	19.31
rRNMPC	0.047	0.078	0.36
LRMPC	$< 10^{-5}$	0.01	0.09

For the same simulation conditions as previously, the robust predictive controllers with penality term on the control variation (6.3) (denoted LRMPC-δu) and the initial formulation (5.33) are compared.

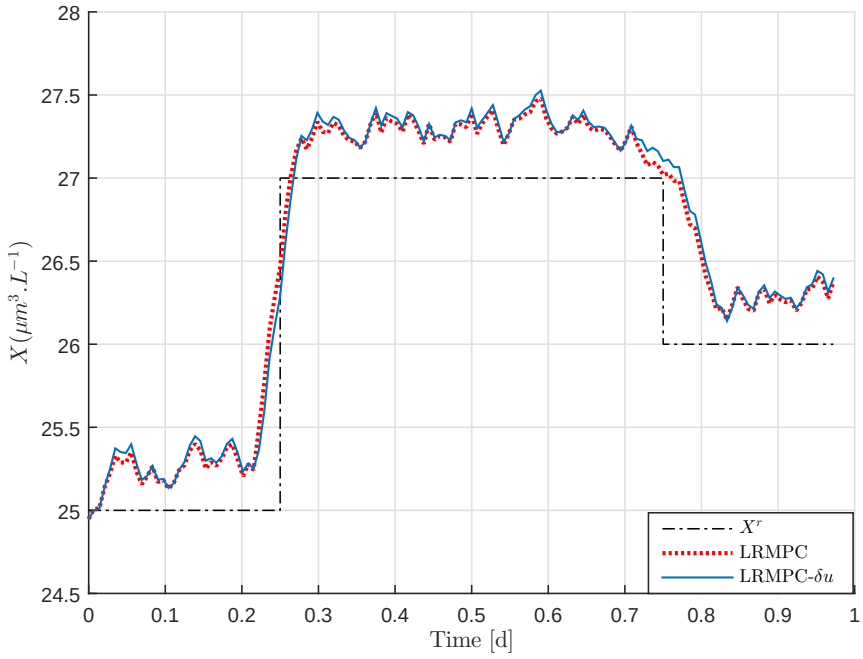

Figure 7.8: Biomass concentration evolution with time for LRMPC-(δu) strategies.

The obtained results in Figure 7.9 (green box) show that the control evolution is slightly smoother when considering control increments (6.3) in comparison with the LRMPC. Thus, the LRMPC insensitivity against measurement noise is improved thanks to the penality term on the control evolution in the cost function.

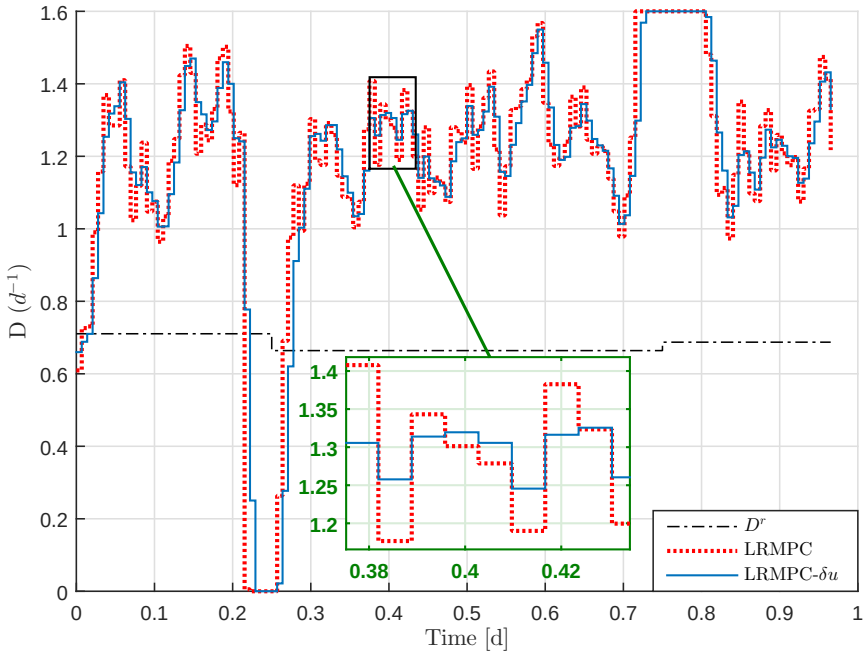

Figure 7.9: Dilution rate evolution with time for LRMPC-(δu) strategies.

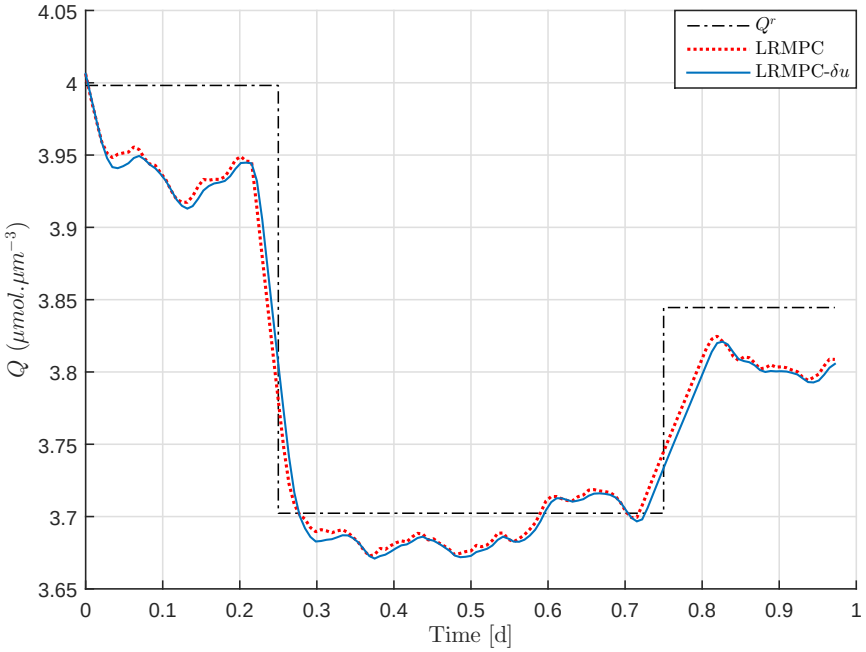

Figure 7.10: Internal quota evolution with time for LRMPC-(δu) strategies.

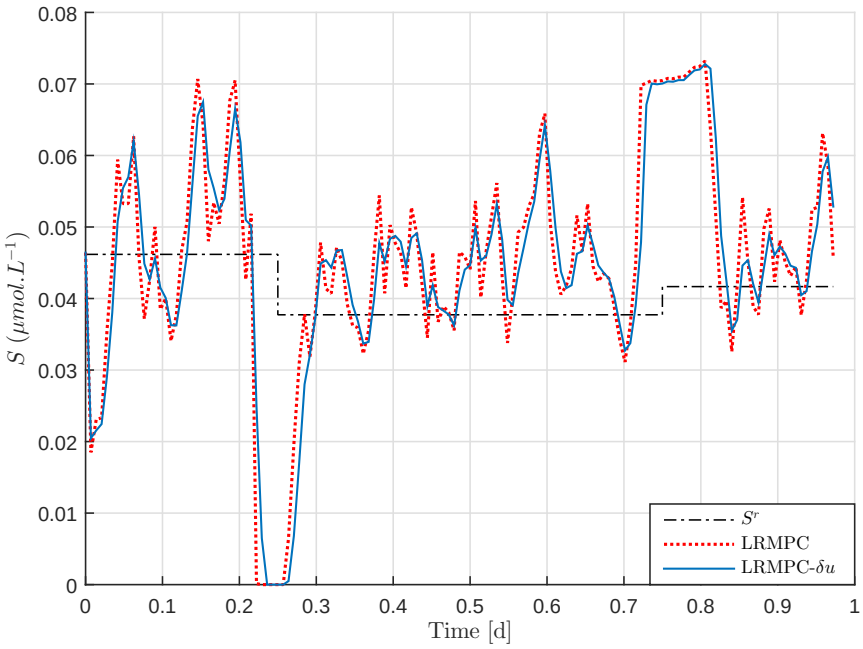

Figure 7.11: Substrate concentration evolution with time for LRMPC-(δu) strategies.

As mentionned previously, the LRMPC or LRMPC-δu accuracy in tracking the setpoint is lower than the one obtained by RNMPC (Figure 7.4). Thus, a hierarchical strategy with a PI and with ISM controllers as detailed in section 6.3.1, are applied.

The PI control input in this case is given by (from (6.35)):

$$\tilde{D}(t) = K_p(X(t) - \hat{X}(t)) + \frac{K_p}{T_i}\int_{t_{k-1}}^{t}(X(\tau) - \hat{X}(\tau))d\tau \tag{7.36}$$

where $\hat{X}$ denotes the biomass model prediction.
The control law derived from the ISM strategy is given by (6.55) (as detailed in section 6.3.2):

$$\begin{aligned}\tilde{D}(t) = -\frac{1}{X(t)}((u_{k-1} - \mu(Q,I))(X(t) - \hat{X}(t)) - \xi_1 K_s \tanh(\phi(x,t)) \\ + (\xi_2 - \xi_1\xi_3)(\xi_1 Z_2(t) + \xi_2 Z_1(t)))\end{aligned} \tag{7.37}$$

Both PI and ISM tuning parameters are determined by a trial-and-error technique (see Table 7.4).

Table 7.4: PI & ISM tuning parameters.

Parameter	K_p	T_i	ξ_1	ξ_2	ξ_3	K_s
Value	1	0.01	1	-1	0.1	1

The obtained results as depicted in Figures 7.12-7.15 compare the LRMPC and hierarchical controller (LRMPC-PI and LRMPC-ISM) performances considering the uncertain parameter worst case cited previously with biomass affected by noise measurement.

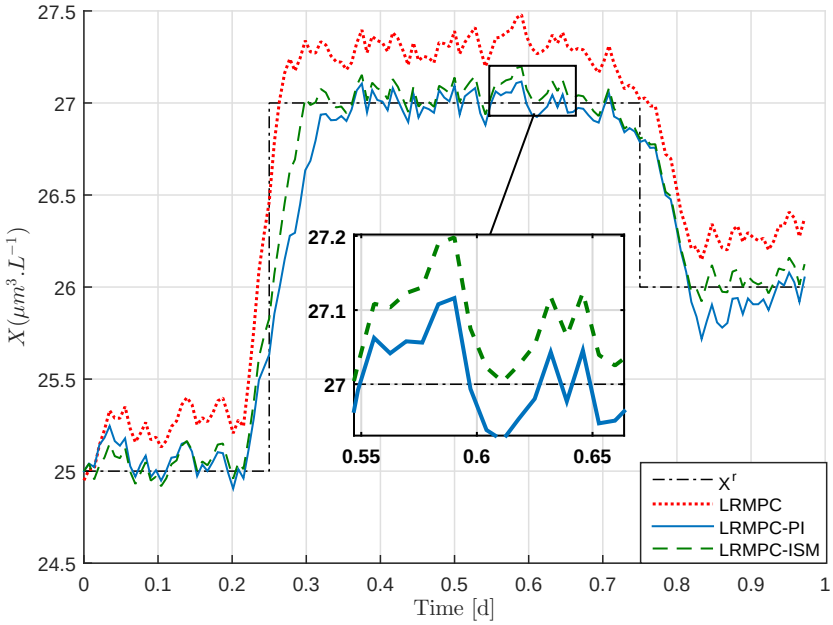

Figure 7.12: Temporal evolution of biomass concentration for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

The use of a hierarchical scheme allows cancelling the static error in comparison with LRMPC law as shown in Figure 7.12 (the obtained accuracy for LRMPC-PI is equivalent to the one obtained with RNMPC in Figure 7.4). The obtained results show that the LRMPC-ISM has slightly better performances than the LRMPC-PI (lower overshoot and lower time response). Nevertheless, in the sequel, the performances analysis will be focused on the LRMPC-PI since it presents the best trade-off between performances and simplicity of implementation as mentionned in section 6.3.1. Figure 7.16 illustrates more spcifically the results obtained by the LRMPC-PI (biomass concentration, and predictive and PI control inputs). It can be observed that the LRMPC-PI control input is non-zero during setpoint change (rising

edge) due to the fact that the PI control is not cancelled. The LRMPC, thanks to it predictive property, cancels the dilution rate so that the growth is maximized, whereas, the PI does not take into account future reference evolution. As a consequence, the time response of the LRMPC-PI is longer than the one achieved by the LRMPC.

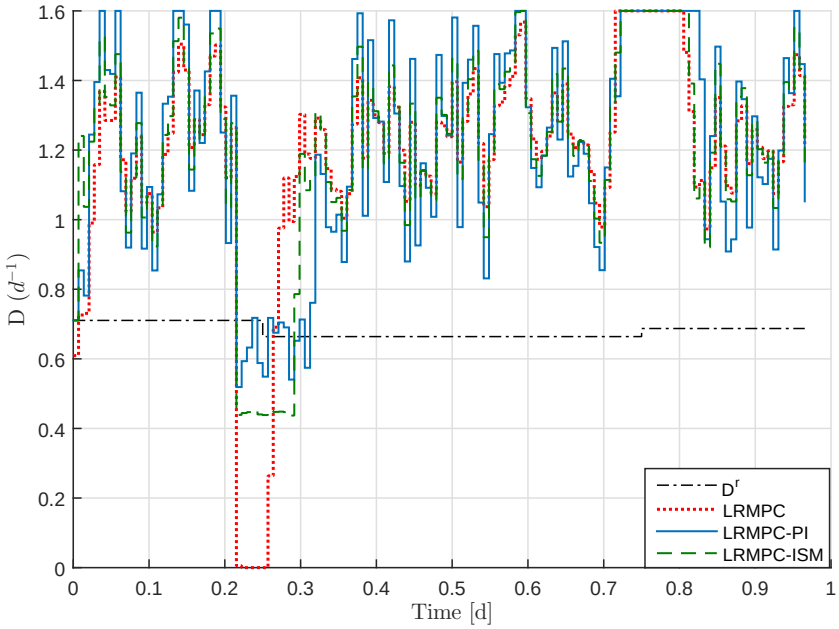

Figure 7.13: Temporal evolution of dilution rate for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

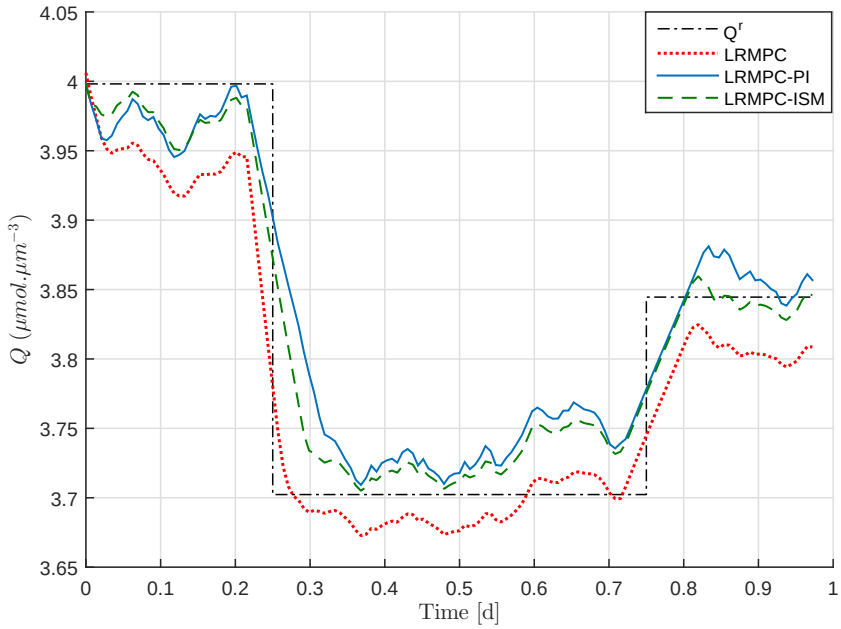

Figure 7.14: Temporal evolution of internal quota for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

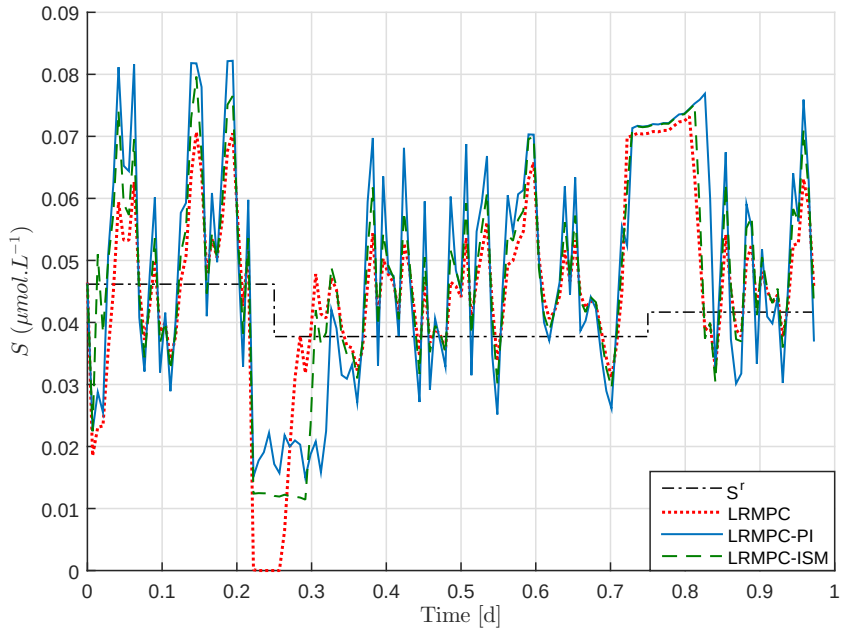

Figure 7.15: Temporal evolution of substrate concentration for LRMPC, LRMPC-PI and LRMPC-ISM strategies.

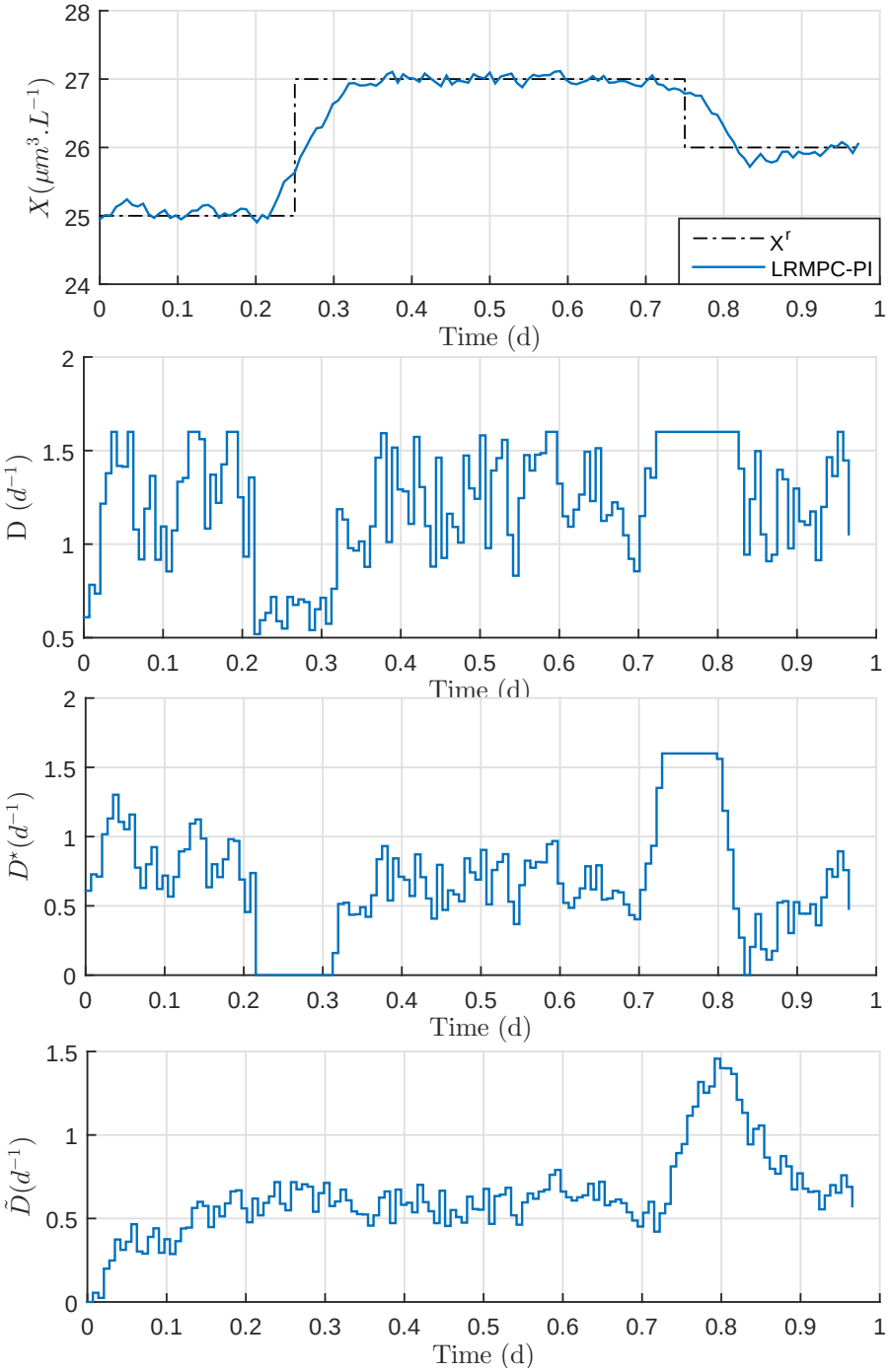

Figure 7.16: Output and control input evolution with time for LRMPC-PI strategy.

A statistical analysis of the robustness is considered in order to highlight the advantage of the hierarchical approach. To emphasize this aspect, simulation for a large number of independent tests are performed with parameters variations for the same conditions as previously. Based on Monte-Carlo procedure, 100 tests have been conducted with a simultaneous random non-correlated variation of 30% at the maximum in all the parameters θ. Figure 7.17 compares NMPC, LRMPC and LRMPC-PI performances. It clearly appears that using the hierarchical algorithm (solid line in blue) allows reducing the standard deviation of tracking error in comparison with a classical NMPC (dotted line in red) and LRMPC (dashed line in green).

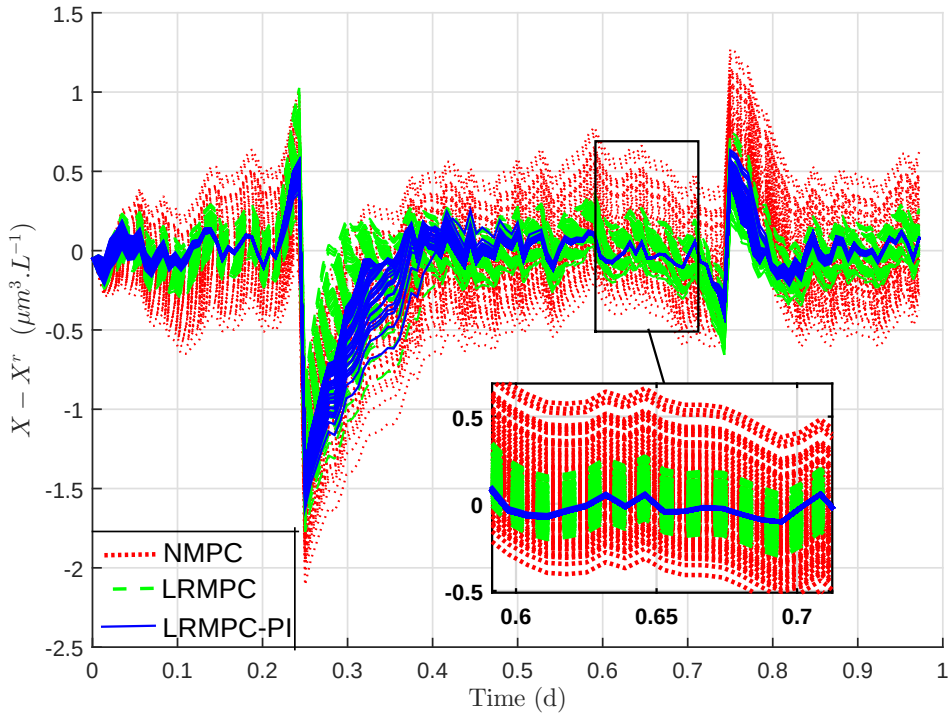

Figure 7.17: Time evolution of tracking error for simultaneous random non correlated variation in all the parameters (Monte-Carlo).

Figure 7.18 shows the resulting histogram of the distribution of the tracking error, and Table 7.5 gives the mean and standard deviation of tracking error for the whole simulation.

These results confirm those obtained in the worst case (Figure 7.12). Furthermore, the results show that the LRMPC has better results than the NMPC in term of accuracy.

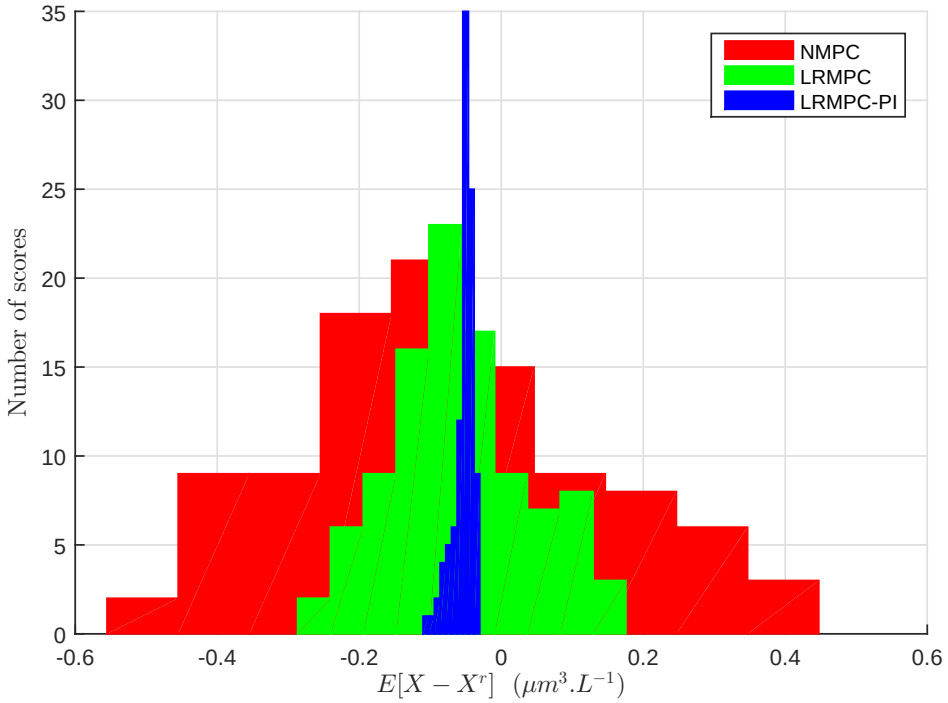

Figure 7.18: Histogram of the average tracking error for simultaneous random non correlated variation in all the parameters (Monte-Carlo).

Table 7.5: Comparison of the proposed algorithms in terms of tracking error distribution features.

Perf. indices / Algo.	mean	standard deviation
NMPC	-0.07	0.219
LRMPC	-0.06	0.098
LRMPC-PI	-0.05	0.014

7.4.3 Reference trajectory tracking

In section 4.4.2, simulations have shown the NMPC performances with $j\varepsilon^{s/m}$ signal for setpoint reference trajectory. In the sequel, a time-varying reference trajectory is selected as shown in Figure 7.19 instead of step-setpoint trajectory. The purpose of this choice is to investigate the NMPC-$j\varepsilon^{s/m}$ (section 4.3) performances in the case of variable reference trajectory to highlight the advantage of the proposed strategy (both LRMPC-PI and LRMPC-ISM). Four controllers will be tested: a classical Nonlinear Model Predictive Control with addition of the error signal $j\varepsilon^{s/m}$ during prediction step (denoted as NMPC-$j\varepsilon^{s/m}$), a robust one using criterion (5.1) (denoted as RNMPC) and two hierarchical control strategies (LRMPC-PI and LRMPC-ISM). Biomass concentration measurements are assumed to be non-corrupted by noise. The light intensity is assumed to be measured and equal to I_{opt}.

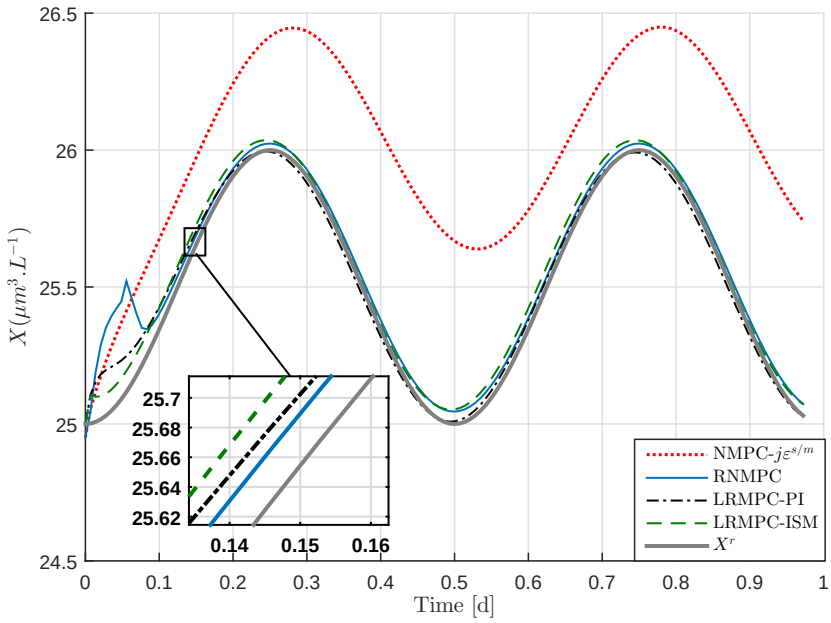

Figure 7.19: Biomass concentration evolution with time for NMPC-$j\varepsilon^{s/m}$, RNMPC, LRMPC-PI and LRMPC-ISM strategies in case of time varying reference trajectory.

It can be seen that including the difference between the system and the model during the prediction in the NMPC law does not improve the tracking accuracy (Figure 7.19). This is due to the fact that the hypothesis on the dynamics of the error between the system and the model, modeled with $j\varepsilon^{s/m}$, is not realistic in the case of time-varying reference trajectory.

On the other hand, both RNMPC and the hierarchical approaches (LRMPC-PI and LRMPC-ISM) have similar performances and lead to a slight dynamic error. The RNMPC leads to the most accurate tracking but presents strong computation time. The above case study highlights the limits to the applicability of the NMPC with $j\varepsilon^{s/m}$ signal.

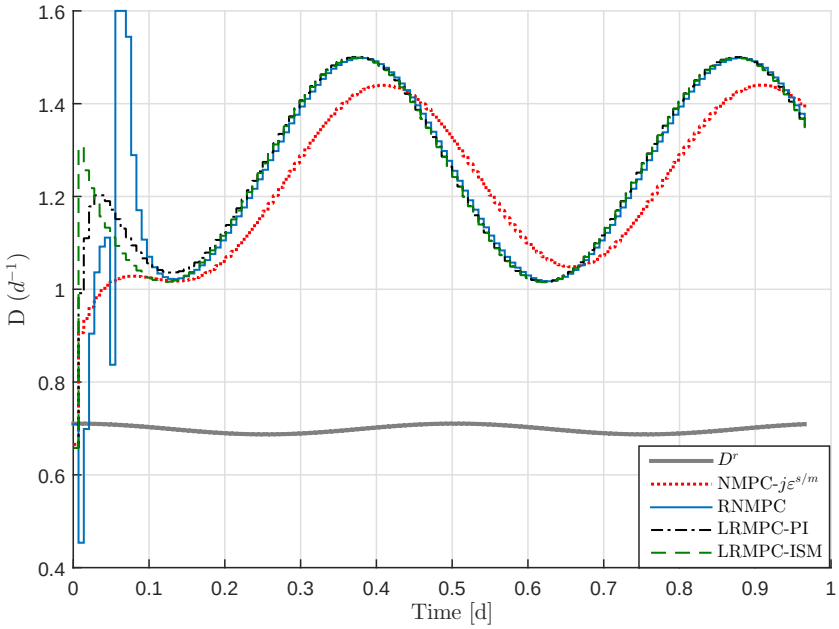

Figure 7.20: Dilution rate evolution with time for NMPC-$j\varepsilon^{s/m}$, RNMPC, LRMPC-PI and LRMPC-ISM strategies in case of time varying reference trajectory.

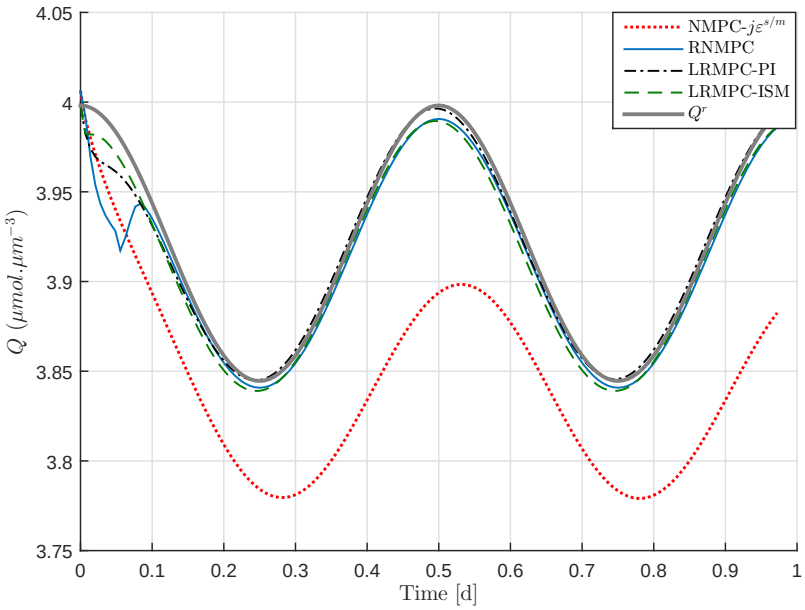

Figure 7.21: Internal quota evolution with time for NMPC-$j\varepsilon^{s/m}$, RNMPC, LRMPC-PI and LRMPC-ISM strategies in case of time varying reference trajectory.

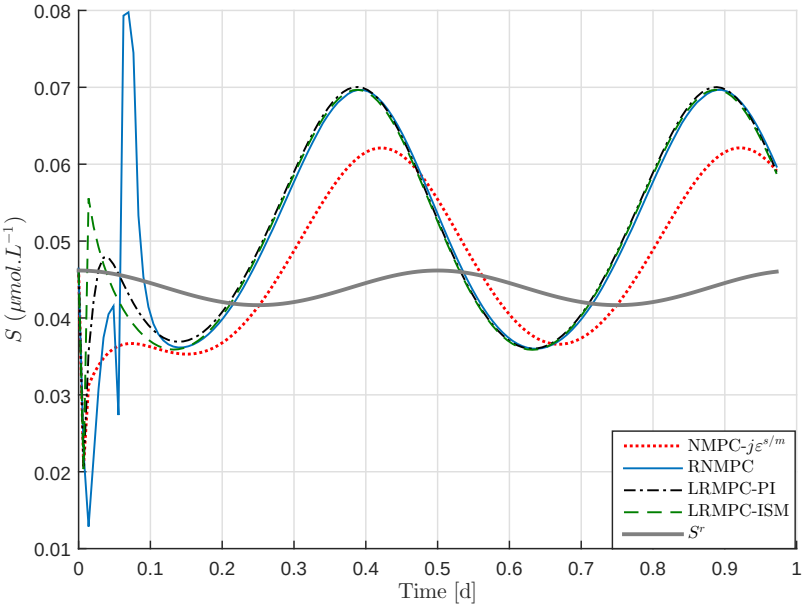

Figure 7.22: Substrate concentration evolution with time for NMPC-$j\varepsilon^{s/m}$, RNMPC, LRMPC-PI and LRMPC-ISM strategies in case of time varying reference trajectory.

7.4.4 Disturbance rejection

The light intensity was set constant in the previous simulations (equal to I_{opt}). In this section, the behavior of the controller in case of time-varying light intensity is discussed. Hereafter, a day/light like variation is considered [144]:

$$I_{real}(t) = \bar{I}(max\{0, sin(2\pi t)\})^2 + Iopt \tag{7.38}$$

where the time t is in days and $\bar{I}$ is set to 280 μE m^{-2} s^{-1}. $I_{nominal} = I_{opt}$ represents the light energy provided by panels to the bioreactor and I_{real} the perturbation modelled as a day/night cycle (*i.e.* non perfectly isolated culture) as shown in Figure 7.25. The biomass concentration setpoint is constant ($X^r = 25\mu m^3/L$). The goal is thus to maintain the biomass concentration as close as possible to this value, despite the light intensity fluctuations. Figures 7.23-7.26 compare the LRMPC, LRMPC-PI and LRMPC-ISM controllers performances. The light intensity considered in the model for prediction is $I_{nominal}$.

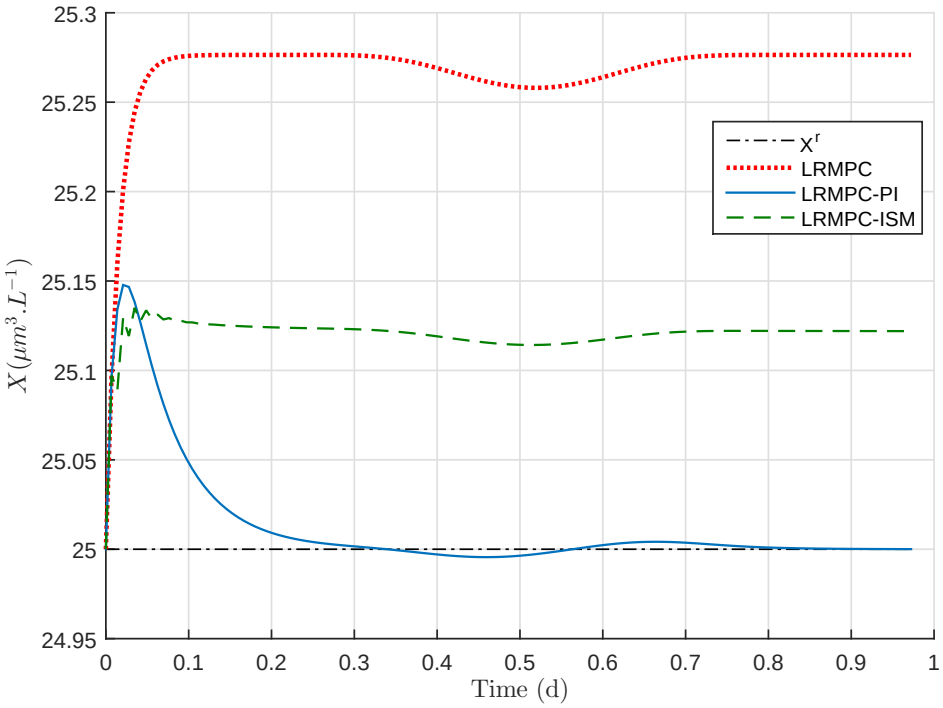

Figure 7.23: Biomass concentration evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies (disturbance rejection).

For both the controllers, when the light I varies between times $t = 0.3$

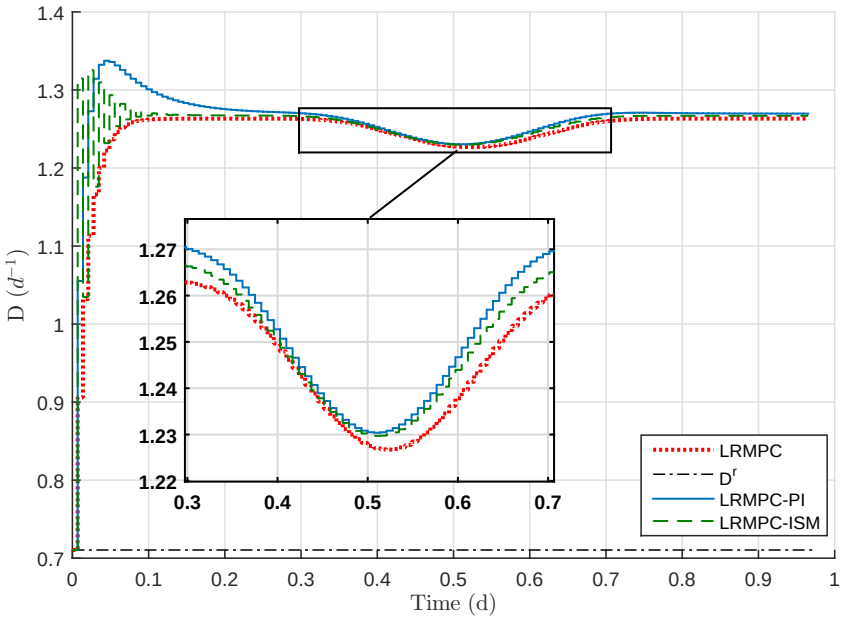

Figure 7.24: Control input evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies (disturbance rejection).

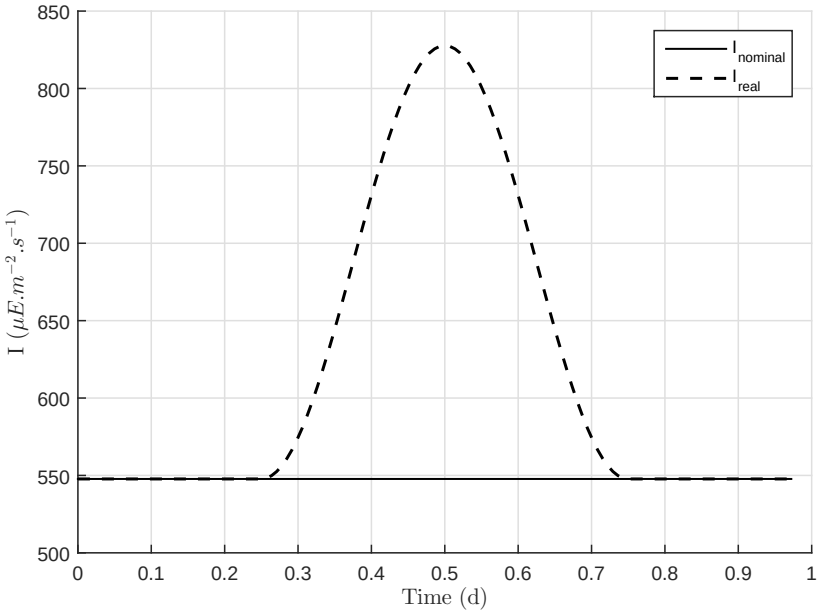

Figure 7.25: Light intensity evolution with time for LRMPC, LRMPC-PI and LRMPC-ISM strategies (disturbance rejection).

and $t = 0.7$ d (Figure 7.25) leading to values higher than the optimal light intensity I_{opt}, and thus, the specific growth rate $\mu(Q, I)$ decreases (inhibition phenomenon as shown in Figure 7.26). As a consequence, the dilution rate D also decreases accordingly in order to maintain the biomass concentration X approximately constant (Figure 7.26) (reminding that D must equals μ to cancel $\dot{X}$). The reference tracking presents an offset in the case of LRMPC as shown in Figure 7.23. On the other side, the hierarchical approach (LRMPC-PI) maintains the biomass concentration at its reference value in the presence of parameters uncertainties and counters the effect of fluctuations in light intensity. In the case of LRMPC-ISM, the non-zero tracking error is due mainly to the use of $\mu(Q, I)$, evaluated at θ_{nom}, in (7.37). This is the main drawback of the LRMPC-ISM. The specific growth rate, μ, could be estimated online, and its estimated value used in the controller.

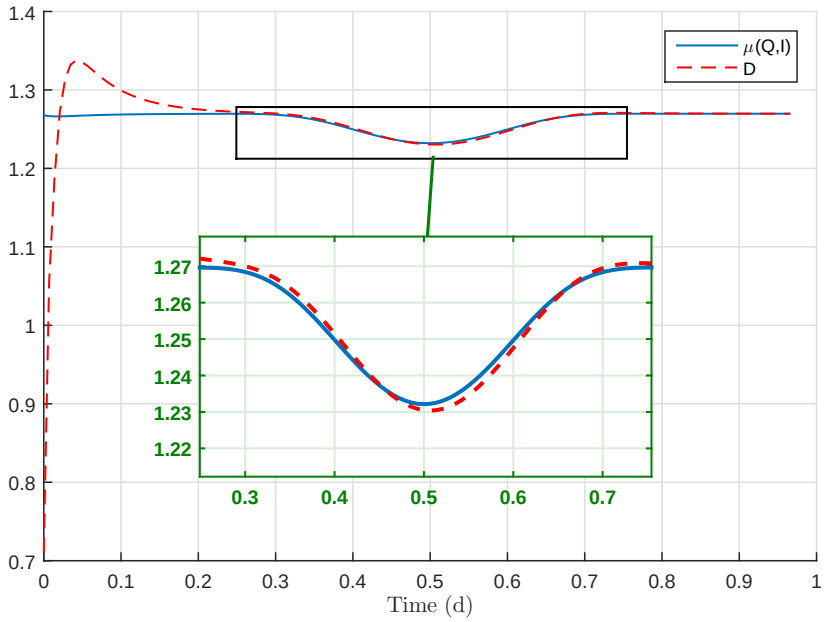

Figure 7.26: Growth rate and dilution rate evolution for LRMPC-PI strategy (disturbance rejection).

7.5 Conclusion

In this chapter, predictive controllers were applied to control microalgae culture in a continuous photobioreactor (Droop model). First, the obtained results show that both RNMPC and LRMPC have better than the classical NMPC performances under parameters uncertainties and measurements affected by noise. The LRMPC accuracy in tracking the setpoint is lower than the one obtained by RNMPC due to the linearization of the prediction model. However, the RNMPC suffers from the difficulty to derive a solution as quickly as possible. The LRMPC offers a strong computational load reduction. Secondly, the use of a hierarchical scheme allows cancelling the static error in comparison with LRMPC law. For a time-varying reference trajectory, the hierarchical control scheme ensures a good tracking accuracy which is not the case for the NMPC-$j\varepsilon^{s/m}$. Thus, it highlights the limits to the applicability of the latter approach. Finally, for time-varying light intensity, the hierarchical approach (LRMPC-PI) maintains the output at its reference value in the presence of parameters uncertainties and counters the effect of fluctuations in light intensity.

This example illustrated and highlighted the advantages of the proposed predictive controllers. More specifically, the hierarchical structure that combines a linearized robust predictive controller with a PI controller proposes a good solution for microalgae cultivation control.

The applicability of this control strategy to other bioprocesses and its possible theoretical and numerical improvements will be discussed in the next chapter.

Chapter 8

General conclusions and future directions

This thesis has investigated new structures of robust nonlinear predictive control, which are able to deal with the compromise between robustness features, tracking accuracy and dynamic performances, on the one hand, and reduced computational load on the other hand. This aspect of computational burden is indeed of major concern, when applying the control strategy in real-time. Even if the bioprocess considered in this work for the application framework has limited real-time constraints, this may not be the case in particular for systems with higher number of uncertain parameters. Therefore the aspect is crucial for RNMPC.

Based on this, the proposed structure, combining a linearized MPC controller with an additional one to deal with residual tracking accuracy, appears to be a good candidate to the paradigm mentioned above.

This development has been elaborated in a progressive way, through the contribution of successive chapters, and assessed through simulations results in the case of bioprocesses.

In this chapter, we summarize the thesis contributions and present recommendations for future work.

8.1 Thesis summary and contributions

Chapter 3 reviewed previous works in the area of predictive control strategies and presents their key advantages and drawbacks. Theoretical, computational and implementation aspect were briefly discussed. It served as a centralized literature study in the thesis.

Chapter 4 made use of the Nonlinear Model Predictive Control (NMPC) discussed in Chapter 3 and appplied it in the context of trajectory tracking problem for continuous/discrete-time nonlinear systems. This strategy induces solving, online, an optimization problem which is expressed as a nonlinear programming problem. This formulation highlighted its limits in the case of plant-model mismatch. One studied improvement consisted to include, during the prediction step, the difference between the system and the model outputs. Unfortunately, the assumption on including the prediction error is quite restrictive due to the fact that we do not know, in a realistic application, how the error will evolve over the prediction horizon.

Chapter 5 focused on the Robust Model Predictive Control. Robust variants of NMPC are able to take into account set bounded uncertainties in the design procedure. In this study, the robust NMPC law was formulated as a min-max optimization problem where the objective function is minimized for the worst possible uncertainty realization. In order to reduce the computation burden, two solutions have been proposed. On one side, thanks to a sensitivity analysis, a reduction of the number of uncertain parameters is performed. Only the main influencial parameters on the model are considered in the min-max problem whereas the other parameters are kept constant and equal to their nominal values. This approach cannot be always used due to two major drawbacks: the necessity to consider a study of the respective impact of the parameters through a sensitivity analysis, leading to a more complex and time demanding study and its applicability only in some cases (when all parameters have high influence on the model, no parameter reduction is possible). On th other side, a new approach based on a model linearization technique (first-order Taylor series expansion) has been developed. It aims to turn the min-max problem into a more tractable optimization problem in order to reduce the computation time as much as possible and to make it suitable for online implementation. The derived optimization problem is an unidimensional optimization (so-called LRMPC). Taking into account the inequality constraints on the optimization variable leads to a bilevel problem (so-called CLRMPC), with a quadratic programming problem in the lower level, and a scalar one in the upper level. This approach leads to a loss of accuracy (linearizaton drawback) but it is computationally tractable in calculating the optimal control compared to the min-max optimization, which makes it suitable for online implementation. Moreover, stability properties of the proposed LRMPC approach has been analyzed.

Chapter 6 proposed two improvements for the developed method. The first

idea deals with the linearization of the model around the nominal parameter values and the optimal control sequence obtained at the previous iteration. This modification led to an optimal control sequence less sensitive ato noise measurements thanks to the inclusion of the penality term on the control evolution. The derived optimization problem is a bilevel one or a scalar minimization, depending on the considered constraints. In a second step, to go further with linearization drawback, a hiearchical control scheme has been proposed. It combines a robust model predictive control law with a proportional Integral (PI) law or Integral Sliding Mode (ISM) controller. The predictive controller guarantees the tracking of the reference trajectory, whereas the additional regulator ensures cancelling any residual tracking error. For the auxiliary controller, the PI law ensures the best trade-off between design complexity and performances, without requiring specific knowledge on the system parameters.

Chapter 7 was devoted to the case study of the control of microalgae culture in continuous photobioreactor. Simulations have been carried out in the case of constant and time-varying reference trajectories to highlight the advantage of the proposed control approach under parameters uncertainities, external perturbation and measurement affected by noise.

8.2 Recommendations for future directions

In this work, the obtained results have highlighted many issues that have to be considered as perspectives of this work. Some recommendations for future work in which further research may be carried out are proposed below, with different time scales.

Short term issues are more related to the application field considered in this work:

- In this dissertation, the proposed control strategies were applied in the case of Single Input Single Output (SISO) system with a small number of uncertain parameters. The application of the proposed methods to a highly nonlinear multivariable system with much more uncertain parameters is under progress (based on the experimental work of [42]). The studied MIMO sytem is a heterotrophic microalgae fed-batch bioreactor. The aim is to control biomass and product concentrations through two substrate feed rates. This objevtive has to be finalized.

- Based on results on the robustness of the LRMPC with respect to model parameters uncertainties, future work can consider extending the study

to the case of disturbances and unmodeled uncertainties (*e.g.* model mismatch of the growth kinetics). In fact, these unmodeled mismatches in the kinetic part could come from the difficulty to select an accurate model structure, a possible evolution of the model parameters and/or a metabolic change over time.

- The biological variables are sometimes not accessible to be measured online and usually measured offline using expensive sensors. In this context, it is important to design an estimation algorithm or so-called soft sensors to rebuild time evolution of the state and to develop a constructive procedure for designing controllers robust against additional estimation errors. Contributions in this area are particularly important.

On top of that, the experimental validation of the proposed robust hierarchical structure on a real bioreactor, either lab-scale or industrial, would be the best validation to fully assess the theoretical developments. This is however a complex task, starting first, for a dedicated benchmark, with the model definition, the parameters identification with their confidence intervals, then the implementation of the control law (coupled to soft sensors) in a dedicated software. This perspective is a really important one to demonstrate the efficiency and advantages of the proposed control strategies.

Long term issues are more related to theoretical developments:

- In order to improve the performances of the LRMPC-δu in the presence of model mismatch and measurements affected by noise (section 6.2.4), an alternative would be to include the penalty term on the control variation (defined in (6.3)) as an additional term in the criterion (5.3) (*i.e.* to consider simultaneously penalty on the control increments and on the difference between u and its reference u^r). We already applied this strategy in the case of NMPC law, leading to good performances (see example in Appendix C). This new criterion should be considered for LRMPC and the control input calculation extended to this new optimization problem.

- In order to increase the quality of the linearized model, there are several issues that deserve further investigation. An interesting perspective may be considering a second order expansion rather than the first order approximation to improve the robustness and accuracy of the LRMPC as in [62].

- In particular, theoretical questions like stability need to be addressed. In fact, stability analysis of the closed-loop system as presented in section 5.4.2 leads to an open question related to the impact of solving the optimization problem by using the Lagrangian duality on the stability of the closed-loop system. Nevertheless, the derived optimization problem being a convex one, leading to better convergence properties than the original min-max problem, an interesting idea is to focus the analysis on the impact of the convergence of the optimization algorithm on the stability of the closed-loop system.

- On the other side, an interesting perspective may be the determination of sufficient conditions ensuring robust stability of the overall hierarchical control scheme, presented in Chapter 6, in case of bounded uncertainties and/or trajectory constraints, starting from the work already done in [131].

- It would be also interesting to analyse the robust stability of the closed loop system (LRMPC strategy coupled with the observer,*e.g.*, Extended Kalman Filter) as done in [69].

Appendix A

A.1 Robust regularized Least squares problem

This part details the development from (5.85) to (5.91) in section 5.5.

Let us consider the Lagrangian given by (5.85):

$$L(z,\xi,\lambda) = -||z||_V^2 - ||Az - b + C\xi||_W^2 + \lambda(||\xi||^2 - \Gamma(z)^2) \tag{A.1}$$

The above equation (A.1) can be expressed as:

$$\begin{aligned} L(z,\xi,\lambda) = &- ||z||_V^2 - \lambda\Gamma(z)^2 - ||Az-b||_W^2 - (Az-b)^\top WC\xi \\ &- \xi^\top C^\top W(Az-b) + \xi^\top(\lambda\mathbb{I} - C^\top WC)\xi \end{aligned} \tag{A.2}$$

The optimal solution $\xi^\star$ that minimizes L is given by:

$$\xi^\star = (\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W(Az-b) \tag{A.3}$$

Replacing (A.3) in (A.2) and after some mathematical manipulationsn, it comes:

$$\begin{aligned} L(z,\xi^\star,\lambda) = &- ||z||_V^2 - \lambda\Gamma(z)^2 - ||Az-b||_W^2 - (Az-b)^\top WC(\lambda\mathbb{I} - C^\top WC)^\dagger \\ &\times C^\top W(Az-b) - (Az-b)^\top WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W(Az-b) \\ &+ (Az-b)^\top WC(\lambda\mathbb{I} - C^\top WC)^\dagger \underbrace{(\lambda\mathbb{I} - C^\top WC)(\lambda\mathbb{I} - C^\top WC)^\dagger}_{\mathbb{I}} \\ &\times C^\top W(Az-b) \end{aligned} \tag{A.4}$$

Then, we have that:

$$\begin{aligned} L(z,\xi^\star,\lambda) = &- ||z||_V^2 - \lambda\Gamma(z)^2 - (Az-b)^\top(W + WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W \\ &\underbrace{+WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W - WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W}_{0})(Az-b) \end{aligned} \tag{A.5}$$

Finally, the Lagrangian (A.1) becomes a function of z and λ as follows:

$$\begin{aligned} L(z,\lambda) = &- ||z||_V^2 - \lambda\Gamma(z)^2 \\ &- (Az-b)^\top \left(W + WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W\right)(Az-b) \end{aligned} \tag{A.6}$$

Let us define $W(\lambda) = W + WC(\lambda\mathbb{I} - C^\top WC)^\dagger C^\top W$.
Then, (A.6) becomes:

$$L(z,\lambda) = -||z||_V^2 - \lambda\Gamma(z)^2 - ||Az-b||_{W(\lambda)}^2 \tag{A.7}$$

which is the expression (5.91).

Appendix B

B.1 Biological systems modelling

This section summarizes general principles in bioprocess modelling. More details can be found in [9]. Biological systems are described by a set of macroscopic components such as biomass, substrates and metabolic products. The mechanistic approach is the most popular bioprocess modelling technique. For optimization, monitoring and control purposes, this approach is usually macroscopic in essence, i.e., it makes use of the concept of macroscopic reaction scheme involving a few reactants, products and catalysts considered as macroscopic entities. The material flow set that defines the transformation of reactants into products is represented in the following form [9]:

$$\sum_{i=1}^{n_s} \Pi_{S_i,k} S_i \quad \overset{r_k(.)}{\rightarrow} \quad \sum_{j=1}^{n_p} \Pi_{P_j,k} P_j, \quad k \in \overline{1, n_r} \tag{B.1}$$

where

- $\Pi_{.,k}$ is the k^{th} pseudo stoichiometric coefficient or yields coefficients. They are negative when they relate to a reactant and positive when they relate to a product.
- $r_k(.)$ is the k^{th} reaction rate.
- The constants n_r, n_s and n_p are respectively the number of reactions, substrates and products.
- S_i and P_j represent respectively the i^{th} component consumed and j^{th} component produced.

Mass balance modelling

Assuming that the reactor is homogeneous, the macroscopic mass balances for each component involved in the reaction network (B.1) lead to a general differential state-space model [9]:

$$\frac{d\Lambda}{dt} = Kr(.) + D(\Lambda_{in} - \Lambda(t)) - G(\Lambda(t)) \tag{B.2}$$

where

- $r(.)$ is the reaction rate vector, (made of $r_k(.)$).
- K the matrix of yields coefficients, (made of $\Pi_{.,k}$).
- Λ the vector of components concentrations.
- Λ_{in} the vector of feed concentrations.
- D the dilution rate which is defined as the medium feed rate over the effective reactor volume.
- $G(\Lambda)$ describes the exchange between the gas and the liquid phases.

B.2 Reaction kinetics modelling

The reaction rates describes how the compenents interact in the reactor based on activation, limitation and inhibition phenomena. The expression of the reaction rates is typically a nonlinear function depending on the components concentrations and some kinetic parameters. As an example, for the reactions associated with a biomass X, the growth reaction is expressed as follows [9]:

$$r_k(.) = \mu_k(.)X \tag{B.3}$$

where μ_k denotes the specific growth rate, which is used to describe the bacterial growth. In the literature, commonly used laws, that describe the reaction kinetics, are determined by the biologists. The most cited one are presented below:

- **Monod model [103]:**

$$\mu = \mu_m \frac{S}{S + K_s} \tag{B.4}$$

  where μ_m is the maximum specifi growth rate (h^{-1}) and K_s is the half saturation constant ($g.L^{-1}$). It models the limitation of the growth by the substrate S.

- **Haldane kinetics [5]:**

$$\mu = \mu_m \frac{S}{S + K_s + \frac{S^2}{K_i}} \tag{B.5}$$

  where K_i is the inhibition constant ($g.L^{-1}$). The growth is inhibited by the substrate S.

- **Cantois kinetics [37]:**

$$\mu = \mu_m \frac{S}{S + XK_c} \tag{B.6}$$

  where K_c is the Contois saturation constant . The growth depends upon the concentrations of both substrate S and biomass X with growth being inhibited at high concentrations of biomass.

These kinetics are also used to model substrate uptake.

In the case of microalgae, the above laws that describe reaction kinetics are used to model both the limitation and the inhibition by the light intensity.

Appendix C

In this section, additional results for the Droop model studied in Chapter 7 are proposed. First, the controllability of model is discussed. Secondly, the dynamics of the sensitivity functions is detailed. Finally, the NMPC-$\varepsilon^{s/m}$ performances are compared to those obtained with the GMC law.

C.1 Controllability

Controllability is concerned with whether one can design control input to steer the state to arbitrary values. Then, in the next, the controllability of the studied system is checked.

The nonlinear system (7.5) is expressed as a control-affine system as follows:

$$\begin{cases} \dot{x} = f_x(x) + f_u(x)u \\ y = h(x) \end{cases} \tag{C.1}$$

with

$$f_x = \begin{bmatrix} \mu(Q,I)X \\ \rho(S) - \mu(Q,I)Q \\ -\rho(S)X \end{bmatrix}, f_u = \begin{bmatrix} -X \\ 0 \\ (S_{in} - S) \end{bmatrix}, \; h(x) = X \tag{C.2}$$

***Theorem* C.1.** The nonlinear system (C.1) is locally controllable at x if and only if the following Lie algebra rank condition is satisfied:

$$\text{rank}(C_o) = n_x \tag{C.3}$$

with

$$C_o = \begin{bmatrix} f_u(x) & \text{ad}_{f_x} f_u(x) & \dots & \text{ad}_{f_x}^{n_x-1} f_u(x) \end{bmatrix} \tag{C.4}$$

Proof. see [73].

***Remark* C.1.** It should be mentionned that there exist several controllability approaches such as controllability, local controllability, weak controllability and local weak controllability. For further details we refer the reader to [68].

Hereafter, condition (C.3) is checked for the studied system.
After developments, we have that

$$\mathrm{ad}_{f_x} f_u(x) = \begin{bmatrix} 0 \\ \dfrac{K_s \bar{\mu} \rho_m (S - S_{in})}{(K_s + S)^2} \\ -\dfrac{K_s \rho_m (s - s_{in}) X}{(K_s + S)^2} \end{bmatrix} \tag{C.5}$$

and

$$\mathrm{ad}^2_{f_x} f_u(x) = \begin{bmatrix} -\frac{K_s \bar{\mu} K_Q \rho_m X (S - S_{in})}{Q^2 (K_s + S)^2} \\ \frac{K_s^2 \bar{\mu}^2 X (S - S_{in}) - K_s \rho_m^2 S X (K_s - S + 2S_{in})}{(K_s + S)^4} + \frac{K_s \bar{\mu} \rho_m (S - S_{in})}{(K_s + S)^2} \\ \frac{K_s \rho_m^2 S X^2 (K_s - S + 2S_{in}) - Ks^2 \rho_m^2 X^2 (S - S_{in})}{(K_s + S)^4} + \frac{K_s \bar{\mu} \rho_m X (\frac{K_Q}{Q} - 1)(S - S_{in})}{(K_s + S)^2} \end{bmatrix} \tag{C.6}$$

Then, the determinent of the controllability matrix C_o is given by:

$$\det(C_o) = \frac{K_s^2 \bar{\mu} K_Q \rho_m^2 X (S - S_{in})^2 (S - S_{in} + QX)}{Q^2 (K_s + S)^4} \tag{C.7}$$

Thus, we obtain

$$\mathbf{det} C_o \neq 0 \ \mathbf{iff} \ \begin{cases} S - S_{in} \neq 0 \\ S - S_{in} + QX \neq 0 \end{cases} \tag{C.8}$$

Based on *Theorem* C.1, the affine-system (C.1)-(C.2) is locally controllable if the condition (C.8) holds.

These conditions are verified for the considered simulation tests.

C.2 Dynamics of sensitivity functions

This part details the development of the dynamics of sensitivity functions for the Droop model (section 7.2).

The dynamics of sensitivities are calculated based on analytical derivation [52] as follows:

$$\frac{d}{dt}\left(\frac{\partial x_i}{\partial \theta_j}\right) = \frac{\partial}{\partial \theta_j}\left(\frac{dx_i}{dt}\right) = \frac{\partial f_i}{\partial \theta_j} + \sum_{k=1}^{n_x} \frac{\partial f_i}{\partial x_k} \frac{\partial x_k}{\partial \theta_j} \tag{C.9}$$

It follows that

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial X}{\partial \theta_j}\right) = \dfrac{\partial \mu}{\partial \theta_j} X + (\mu(Q,I) - D)\dfrac{\partial X}{\partial \theta_j} + \dfrac{\partial \mu(Q,I)}{\partial Q}\dfrac{\partial Q}{\partial \theta_j} X \\ \dfrac{d}{dt}\left(\dfrac{\partial Q}{\partial \theta_j}\right) = \dfrac{\partial \rho(S)}{\partial \theta_j} - \dfrac{\partial \mu(Q,I)}{\partial \theta_j} Q - \mu(Q,I)\dfrac{\partial Q}{\partial \theta_j} + \dfrac{\partial \rho(S)}{\partial S}\dfrac{\partial S}{\partial \theta_j} - \dfrac{\partial \mu(Q,I)}{\partial Q}\dfrac{\partial Q}{\partial \theta_j} Q \\ \dfrac{d}{dt}\left(\dfrac{\partial S}{\partial \theta_j}\right) = -\dfrac{\partial S}{\partial \theta_j} D - \dfrac{\partial \rho(S)}{\partial \theta_j} X - \rho(S)\dfrac{\partial X}{\partial \theta_j} - \dfrac{\partial \rho(S)}{\partial S}\dfrac{\partial S}{\partial \theta_j} X \end{cases} \tag{C.10}$$

with

$$\frac{\partial \mu(Q,I)}{\partial \theta^\top} = \begin{bmatrix} 0 \\ 0 \\ \left(1 - \frac{K_Q}{Q}\right) \frac{I}{I + K_{sI} + \frac{I^2}{K_{iI}}} \\ -\frac{\bar{\mu}}{Q} \frac{I}{I + K_{sI} + \frac{I^2}{K_{iI}}} \\ -\bar{\mu}\left(1 - \frac{K_Q}{Q}\right) \frac{I}{(I + K_{sI} + \frac{I^2}{K_{iI}})^2} \\ \bar{\mu}\left(1 - \frac{K_Q}{Q}\right) \frac{I^3}{K_{iI}^2 (I + K_{sI} + \frac{I^2}{K_{iI}})^2} \\ 0 \end{bmatrix}, \frac{\partial \rho(S)}{\partial \theta^\top} = \begin{bmatrix} \frac{S}{S+K_s}\left(1 - \frac{Q}{Q_l}\right) \\ -\rho_m \left(1 - \frac{Q}{Q_l}\right) \frac{S}{(S+K_s)^2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho_m \frac{SQ}{S+K_s} \frac{1}{Q_l^2} \end{bmatrix}$$

and

$$\frac{\partial \mu(Q,I)}{\partial x^\top} = \begin{bmatrix} 0 \\ \bar{\mu} \dfrac{I}{I + K_{sI} + \dfrac{I^2}{K_{iI}}} \dfrac{K_Q}{Q^2} \\ 0 \end{bmatrix}, \quad \frac{\partial \rho(S)}{\partial x^\top} = \begin{bmatrix} 0 \\ -\dfrac{\rho_m}{Q_l} \dfrac{S}{S + K_s} \\ \rho_m \left(1 - \dfrac{Q}{Q_l}\right) \dfrac{K_s}{(S + K_s)^2} \end{bmatrix}$$

Finally, it comes:

$$
\begin{cases}
\frac{d}{dt}\left(\frac{\partial X}{\partial \rho_m}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial \rho_m} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial \rho_m} + \frac{\partial \mu(Q,I)}{\partial \rho_m}X \\
\frac{d}{dt}\left(\frac{\partial X}{\partial K_s}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial K_s} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial K_s} + \frac{\partial \mu(Q,I)}{\partial K_s}X \\
\frac{d}{dt}\left(\frac{\partial X}{\partial \bar{\mu}}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial \bar{\mu}} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial \bar{\mu}} + \frac{\partial \mu(Q,I)}{\partial \bar{\mu}}X \\
\frac{d}{dt}\left(\frac{\partial X}{\partial K_Q}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial K_Q} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial K_Q} + \frac{\partial \mu(Q,I)}{\partial K_Q}X \\
\frac{d}{dt}\left(\frac{\partial X}{\partial K_{sI}}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial K_{sI}} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial K_{sI}} + \frac{\partial \mu(Q,I)}{\partial K_{sI}}X \\
\frac{d}{dt}\left(\frac{\partial X}{\partial K_{iI}}\right) = (\mu(Q,I)-D)\frac{\partial X}{\partial K_{iI}} + \frac{\partial \mu(Q,I)}{\partial Q}X\frac{\partial Q}{\partial K_{iI}} + \frac{\partial \mu(Q,I)}{\partial K_{iI}}X \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial \rho_m}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial \rho_m} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial \rho_m} + \frac{\partial \rho(S)}{\partial \rho_m} - \frac{\partial \mu(Q,I)}{\partial \rho_m}Q \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial K_s}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial K_s} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial K_s} + \frac{\partial \rho(S)}{\partial K_s} - \frac{\partial \mu(Q,I)}{\partial K_s}Q \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial \bar{\mu}}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial \bar{\mu}} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial \bar{\mu}} + \frac{\partial \rho(S)}{\partial \bar{\mu}} - \frac{\partial \mu(Q,I)}{\partial \bar{\mu}}Q \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial K_Q}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial K_Q} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial K_Q} + \frac{\partial \rho(S)}{\partial K_Q} - \frac{\partial \mu(Q,I)}{\partial K_Q}Q \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial K_{sI}}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial K_{sI}} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial K_{sI}} + \frac{\partial \rho(S)}{\partial K_{sI}} - \frac{\partial \mu(Q,I)}{\partial K_{sI}}Q \\
\frac{d}{dt}\left(\frac{\partial Q}{\partial K_{iI}}\right) = -(\mu(Q,I)+\frac{\partial \mu(Q,I)}{\partial Q})\frac{\partial Q}{\partial K_{iI}} + \frac{\partial \rho(S)}{\partial S} + \frac{\partial S}{\partial K_{iI}} + \frac{\partial \rho(S)}{\partial K_{iI}} - \frac{\partial \mu(Q,I)}{\partial K_{iI}}Q \\
\frac{d}{dt}\left(\frac{\partial S}{\partial \rho_m}\right) = -\rho(S)\frac{\partial X}{\partial \rho_m} - (D + \frac{\partial \rho(S)}{\partial S}X)\frac{\partial S}{\partial \rho_m} - \frac{\partial \rho(S)}{\partial \rho_m}X \\
\frac{d}{dt}\left(\frac{\partial S}{\partial K_s}\right) = -\rho(S)\frac{\partial X}{\partial K_s} - (D + \frac{\partial \rho}{\partial S}X)\frac{\partial S}{\partial K_s} - \frac{\partial \rho(S)}{\partial K_s}X \\
\frac{d}{dt}\left(\frac{\partial S}{\partial \bar{\mu}}\right) = -\rho(S)\frac{\partial X}{\partial \bar{\mu}} - (D + \frac{\partial \rho(S)}{\partial S}X)\frac{\partial S}{\partial \bar{\mu}} - \frac{\partial \rho(S)}{\partial \bar{\mu}}X \\
\frac{d}{dt}\left(\frac{\partial S}{\partial K_Q}\right) = -\rho(S)\frac{\partial X}{\partial K_Q} - (D + \frac{\partial \rho(S)}{\partial S}X)\frac{\partial S}{\partial K_Q} - \frac{\partial \rho(S)}{\partial K_Q}X \\
\frac{d}{dt}\left(\frac{\partial S}{\partial K_{sI}}\right) = -\rho(S)\frac{\partial X}{\partial K_{sI}} - (D + \frac{\partial \rho(S)}{\partial S}X)\frac{\partial S}{\partial K_{sI}} - \frac{\partial \rho(S)}{\partial K_{sI}}X \\
\frac{d}{dt}\left(\frac{\partial S}{\partial K_{iI}}\right) = -\rho(S)\frac{\partial X}{\partial K_{iI}} - (D + \frac{\partial \rho(S)}{\partial S}X)\frac{\partial S}{\partial K_{iI}} - \frac{\partial \rho(S)}{\partial K_{iI}}X
\end{cases}
$$

C.3 Sensitivity analysis

Hereafter, a sensitivity analysis is carried out to determine the most influencial parameters. To simplify the study of the sensitivity functions, each function is rescaled by multiplying it by the parameter value under study. The state x_i is said to be sensitive to a given parameter if a change in the parameter's value significantly affects the predictive quality of the model. The sensitivity of one state with respect to all parameters is to be considered separately and the different orders of magnitude have to be compared. For the considered system (i.e. 3 state variables and 6 parameters), 3×6 sensitivity functions are then computed for biomass concentration $X = 25\mu\text{m}^3\,\text{L}^{-1} > X_l$ (satisfying (7.35)). The corresponding dilution rate is calculated at the equilibrium using (7.33). The sensitivity analysis according to the magnitude order of the sensitivity functions allows to select the parameters which are significantly the most influential.
According to results obtained in Figure C.1, the model parameters are listed in order of descending influence in the Table C.1. The same procedure is applied for S and Q (data not shown). The substrate half saturation constant K_s and the minimal cell quota K_Q are the most influential parameters

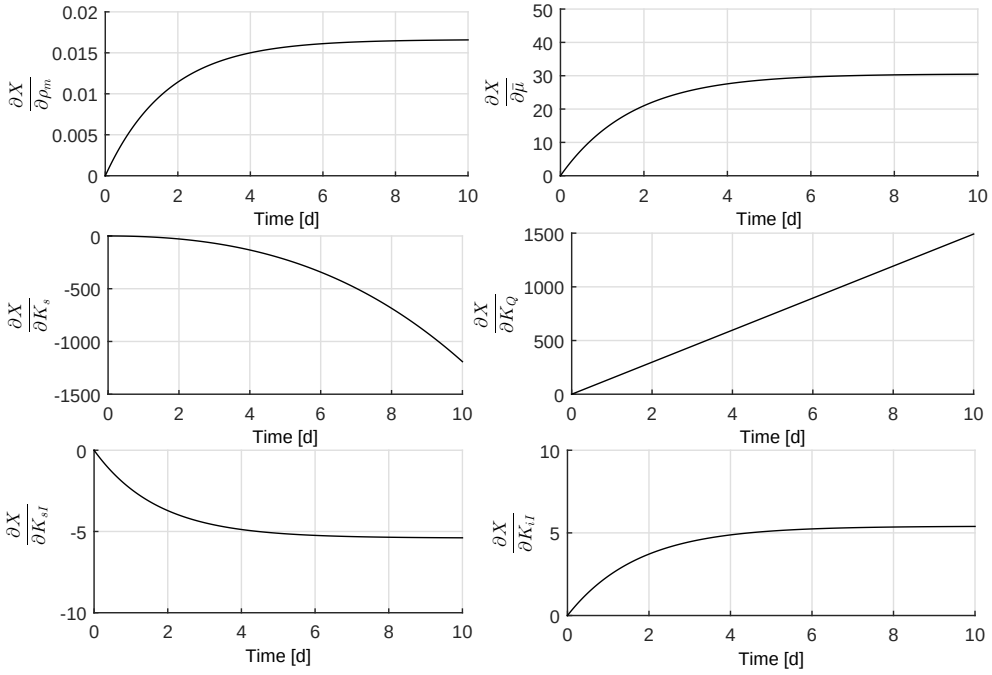

Figure C.1: Evolution of the scaled sensitivity functions (for biomass concentration).

on the biomass concentration evolution (100 times higher than the other parameters).

Table C.1: The ranking of parameters according to their influence on the model (from more to less).

X	Q	S
K_Q	K_Q	K_Q
K_s	K_s	K_s
$\bar{\mu}$	$\bar{\mu}$	ρ_m
K_{sI}	K_{sI}	$\bar{\mu}$
K_{iI}	K_{iI}	K_{sI}
ρ_m	ρ_m	K_{iI}

Thereafter, only these two parameters (K_s,K_Q) are considered in the min-max optimization problem, instead of the 6 model parameters (the 4 other parameters are set to their nominal values).

C.4 Additional simulation results

In this section, the NMPC-$\varepsilon^{s/m}$ and Generic Model Control (GMC) performances for reference tracking are compared in the case of model mismatch and measurement noise.

C.4.1 Predictive controller

The formulation of the NMPC-$\varepsilon^{s/m}$ optimization problem at each sampling instant kT_s is as follows:

$$\begin{aligned} \overset{\star}{D}_k^{\,k+N_p-1} = \arg \min_{D_k^{k+N_p-1}} \quad & \sum_{j=1}^{N_p} \Big((X^r_{k+j} - \hat{X}_{k+j|k})^2 \\ & + \lambda (D^r_{k+j-1} - D_{k+j-1})^2 \Big) \\ & + \beta \sum_{j=1}^{N_p-1} (D_{k+j} - D_{k+j-1})^2 \end{aligned} \tag{C.11}$$

subject to

$$\begin{cases} \hat{X}_{k+j|k} = Hg(t_k, t_{k+j}, x_k, D_k^{k+j-1}, \theta_{\text{nom}}), \quad j = \overline{1, N_p} \\ D_k \geq 0, \quad X_k \geq 0, \quad \forall k \in \mathbb{N} \end{cases} \tag{C.12}$$

where λ is the control weighting factor ensuring that D remains close to its reference D^r, β is a penalty weighting factor on the control variation term and

$\hat{X}_{k+j|k}$ is the predicted biomass concentration. Model uncertainties are taken into account by assuming that the gap between the system and the model at sample time k, denoted $\varepsilon^{s/m}$, is included in the prediction as detailed in section 4.3.

C.4.2 Generic model control

The GMC is a kind of Nonlinear Proportional Integral regulator. It includes proportional and integral actions, cancelling static errors, and performs an input/output linearization of the system. Details related to the GMC law can be found in [82]. In the case of the considered control problem, the dilution rate derived from the GMC law is given by:

$$D = -\frac{1}{X}\left(G_1(X^r - X) + G_2\int_0^t (X^r - X)d\tau - \mu(Q, I)X\right) \tag{C.13}$$

The GMC tuning parameters ($G_1 = 2\xi\omega_0$ and $G_2 = \omega_0^2$) are chosen based on the fact that a second order behavior is specified for the closed-loop system. The damping factor ξ gives a desired shape of response, and the natural frequency ω_0 an appropriate settling time.

C.4.3 Simulation results

The initial biomass concentration value is set close to the setpoint in order to cancel the transient effect and to focus only on the behavior during setpoint changes. The simulation time is fixed at 3 d, the sample time T_s is chosen equal to 30 min. Only uncertainties on the first four parameters of vector θ are considered. In this case, the four first parameters of the mismatched model are chosen on the parameter subspace border ($\theta_{real} = [\theta_1^+, \theta_2^-, \theta_3^+, \theta_4^-]$), where the uncertain parameters subspace $[\theta^-, \theta^+]$ is given by $[\frac{\theta_{\text{nom}}}{2}, \frac{3\theta_{\text{nom}}}{2}]$. The two last parameters K_{sI} and K_{iI} are set to their nominal values. Biomass concentration measurements, y_k, are assumed to be corrupted by a Gaussian white noise of zero mean and 0.05 standard deviation. The light intensity is assumed to be measured online, non-corrupted with noise.

The tuning parameters of the NMPC are determined by a trial-and-error technique. The prediction horizon of the NMPC law is set to N_p=10, chosen to satisfy a compromise between the computation time and a sufficient vision of the system behaviour in the future. The weighting factor λ is selected at 1 to promote the tracking error on biomass concentration relatively to the tracking error on dilution rate in the objective function. In addition, the penalty weighting factor on the control variation β is chosen equal to 10λ.

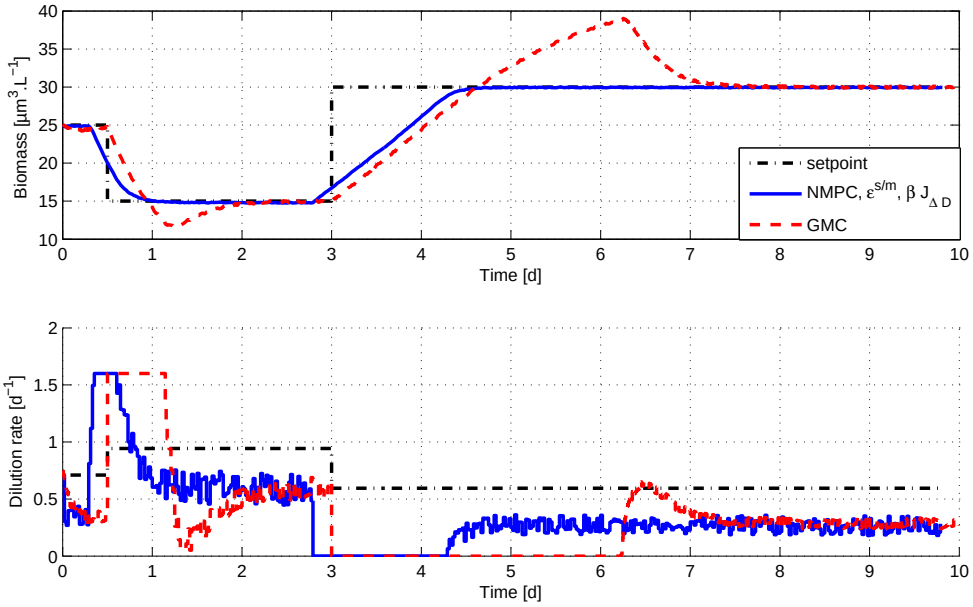

Figure C.2: Evolution of biomass concentration and dilution rate with measurement affected by noise, GMC and NMPC-$\varepsilon^{s/m}$ laws.

This penalization term was chosen to allow a good trade-off between noise rejection and trajectory tracking. The GMC tuning parameters are chosen as follows: ξ =1 and ω_0 =5 rad d^{-1}.
For the tuning parameters ξ =1 and ω_0 =5 rad d^{-1}, it can be observed (Figure C.2) that the peak time is well respected. The GMC law leads to a larger overshoot in biomass concentration evolution during a setpoint change, with a longer transient and response time in comparison with the NMPC performances (mainly due to the lack of anticipation). In the case of the NMPC, it can be noticed the anticipation of a setpoint change due to the prediction of the model behavior in the moving horizon.

More details on this example can be found in [14].

Bibliography

[1] J. Abdollahi and S. Dubljevic. Lipid production optimization and optimal control of heterotrophic microalgae fed-batch bioreactor. *Chem. Eng. Sci*, 84:619–627, 2012.

[2] M. Alamir and G. Bornard. Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. *Automatica*, 31(9):1353–1356, 1995.

[3] F. Allgöwer and A. Zheng. *Nonlinear Model Predictive Control*. Progress in Systems and Control Theory. Birkhäuser Basel, 2012.

[4] J.C. Allwright. *Advances in Model-Based Predictive Control*, chapter on min-max Model-Based Predictive Control. Oxford University Press, 1994.

[5] J.F. Andrews. A mathematical model for continuous culture of microorganism utilizing inhibitory substrate. *Biotechnology and Bioengineering*, 10:707 – 723, 1968.

[6] D. Angeli, A. Casavola, and E. Mosca. Constrained predictive control of nonlinear plants via polytopic linear system embedding. *International Journal of Robust and Nonlinear Control*, 10(13):1091–1103, 2000.

[7] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, and M. R. Bakhtiari. Optimal control of a nonlinear fed-batch fermentation process using model predictive approach. *Journal of Process Control*, 9:1162–1173, 2009.

[8] J. G. Balchen, D. Ljungquist, and S. Strand. State space model predictive control of a multi stage electro-metallurgical process. *Modeling, Identification and Control*, 10(1):35–51, 1989.

[9] G. Bastin and D. Dochain. *On-line estimation and adaptive control of bioreactors*. Elsevier, 1990.

[10] G. Becerra-Celis, G. Hafidi, S. Tebbani, and D. Dumur. Nonlinear predictive control for continuous microalgae cultivation process in a photobioreactor. *Proc. of the 10th ICARV Conference*, pages 1373–1378, 2008.

[11] G. Becerra-Celis, S. Tebbani, C. Joannis-Cassan, A. Isambert, and H. Siguerdidjane. Control strategy for continuous microalgae cultivation process in a photobioreactor. *Proc. of the 17th IEEE CCA*, pages 684–689, 2008.

[12] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515, 11 1954.

[13] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002.

[14] S. E. Benattia, S. Tebbani, and D. Dumur. Nonlinear model predictive control for regulation of microalgae culture in a continuous photobioreactor. *Proc. of the 22nd MED Conference*, pages 469–474, 2014.

[15] O. Bernard. Hurdles and challenges for modelling and control of microalgae for CO_2 mitigation and biofuel production. *J. of Process Control*, 21(10):1378–1389, 2011.

[16] O. Bernard and J-L. Gouzé. Transient behavior of biological loop models, with application to the droop model. *Mathematical Biosciences*, 127, 1995.

[17] L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8(3–4):243–247, 1984.

[18] L.T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, 46(11):1043 – 1053, 2007.

[19] L.T. Biegler, X. Yang, and G.A.G Fischer. Advances in sensitivity-based nonlinear model predictive control and dynamic real time optimization. *J. of Process Control*, 30:104–116, 2015.

[20] R. Bitmead, V. Wertz, and M. Gevers. *Adaptive Optimal Control: The Thinking Man's GPC*. Prentice Hall Professional Technical Reference, 1991.

[21] E. Bourgeois, S. Tebbani, and A. Ramos Espinosa. Launcher atmospheric guidance based on nonlinear model predictive control. In *GNC 2014*, pages CD–Rom, 2014.

[22] S. Boussiba and S. Leu. Microalgal biotechnology for environmental remediation. *7th European Workshop Biotechnology of Microalgae*, 2007.

[23] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[24] F. Breitenecker, A. Kugi, I. Troch, M. Benavides, D. Telen, J. Lauwers, F. Logist, J. Van Impe, and A. Vande Wouwer. Parameter identification of the droop model using optimal experiment design. *IFAC-PapersOnLine 8th Vienna International Conference on Mathematical Modelling*, 48(1):586 – 591, 2015.

[25] A.E. Bryson and Y.C. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Halsted Press book. Taylor & Francis, 1975.

[26] C. Büskens and H. Maurer. *Online Optimization of Large Scale Systems*, chapter Sensitivity Analysis and Real-Time Control of Parametric Optimal Control Problems Using Nonlinear Programming Methods, pages 57–68. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[27] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer London, 2004.

[28] M. Cannon, D. Ng, and B. Kouvaritakis. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter Successive Linearization NMPC for a Class of Stochastic Nonlinear Systems, pages 249–262. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[29] A. Casavola, D. Famularo, and G. Franze. Predictive control of constrained nonlinear systems via LPV linear embedding. *International Journal of Robust and Nonlinear Control*, 13:281–294, 2003.

[30] B. Chachuat, B. Srinivasan, and D. Bonvin. Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, 33(10):1557 – 1567, 2009.

[31] C.C. Chen and L. Shaw. On receding horizon feedback control. *Automatica*, 18(3):349 – 352, 1982.

[32] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[33] Y. Chisti. Biodiesel from microalgae. *Biotechnology Advances*, 25:294–306, 2007.

[34] P.D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21 – 41, 2013.

[35] D. W. Clarke. Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine*, 8(2):49–55, 1988.

[36] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control-Part I. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[37] D.E. Contois. Kinetics of bacterial growth: Relationship between population density and specific growth rate of continuous cultures. *Journal of General Microbiology*, 21:40 – 50, 1959.

[38] P. Cougnon, D. Dochain, M. Guay, and M. Perrier. Online optimization of fedbatch bioreactors by adaptive extremum seeking control. *J. of Process Control*, 21:1526–1532, 2011.

[39] J. Crassidis, F. Markley, T. C Anthony, and S. Andrews. Nonlinear predictive control of spacecraft. *Journal of Guidance, Control, and Dynamics*, 20(6):1096–1103, 1997.

[40] C.R. Cutler and B.C. Ramaker. Dynamic matrix control- a computer control algorithm. In *Automatic Control Conference*, 1980.

[41] R.M.C. De keyser and A.R. Van Cauwenberghe. Extended prediction self-adaptive control. In *In IFAC Symp. on Identification and System Parameter Estimation*, pages 1317–1322, 1985.

[42] H. De la Hoz Siegler. *Optimization of biomass and lipid production in heterotrophic microalgal cultures*. University of Alberta, 2012.

[43] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing recedin-horizon control of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998.

[44] N. M.C. De Oliveira and L. T. Biegler. An extension of Newton-type algorithms for nonlinear process control. *Automatica*, 31(2):281 – 286, 1995.

[45] D. DeHaan and M. Guay. A new real-time perspective on non-linear model predictive control. *Journal of Process Control*, 16(6):615 – 624, 2006.

[46] P. Deuflhard. A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numerische Mathematik*, 22(4):289–315.

[47] L. Dewasme, S. Fernandes, Z. Amribt, L.O. Santos, Ph. Bogaerts, and A. Vande Wouwer. State estimation and predictive control of fed-batch cultures of hybridoma cells. *Journal of Process Control*, 30:50 – 57, 2015.

[48] C. Diaz, P. Dieu, C. Feuillerat, P. Lelong, and M. Salome. Adaptive predictive control of dissolved oxygen concentration in a laboratory-scale bioreactor. *J. of Biotechnol.*, 43:21–32, 1995.

[49] M. Diehl, R. Amrit, and J. B. Rawlings. A lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56(3):703–707, 2011.

[50] M. Diehl, H. G. Bock, and J. P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[51] M. Diehl, H. J. Ferreau, and N. Haverbeke. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pages 391–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[52] D. Dochain. *Automatic control of bioprocesses*. Editor. John Wiley & Sons, 2008.

[53] D. Dochain and M. Perrier. Adaptive backstepping nonlinear control of bioprocesses. *7th International Symposium, Advanced control of chemical processes*, pages 77–82, 2004.

[54] D.Z. Du and P.M. Pardalos. *Minimax and Applications*. Nonconvex Optimization and Its Applications. Springer US, 2013.

[55] C. Enzing, M. Ploeg, M. Barbosa, and L. Sijtsma. Microalgae-based products for the food and feed sector: an outlook for Europe. Technical report, European comission, 2014.

[56] C.D.M. Filipe and C.P.L. Grady Jr. *Biological Wastwater Treatment*. Marcel Dekker, 1998.

[57] R. Findeisen, L. Imsland, F. Allgöwer, and B.A. Foss. State and output feedback nonlinear model predictive control: An overview. *European Journal of Control*, 9(2):190 – 206, 2003.

[58] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice-a survey. *Automatica*, 25(3):335 – 348, 1989.

[59] C.E. Garcia and M. Morari. Internal model control. a.

[60] A. Gautam, Y. C. Chu, and Y. C. Soh. Robust H_∞ receding horizon control for a class of coordinated control problems involving dynamically decoupled subsystems. *IEEE Transactions on Automatic Control*, 59(1):134–149, 2014.

[61] G. Goffaux and A. Vande Wouwer. Design of a robust nonlinear receding-horizon observer-Application to a biological system. *Proc. of the 17th IFAC World Congress*, pages 15553–15558, 2008.

[62] G. Goffaux and A. Vande Wouwer. *Design of a Robust Nonlinear Receding-Horizon Observer - First-Order and Second-Order Approximations*, pages 295–304. 2009.

[63] G. C. Goodwin, J. Østergaard, D. E. Quevedo, and A. Feuer. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter A Vector Quantization Approach to Scenario Generation for Stochastic NMPC, pages 235–248. Springer Berlin Heidelberg, 2009.

[64] A. Grancharova, T. A. Johansen, and P. Tøndel. *Explicit Nonlinear Model Predictive Control*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[65] E. Gyurkovics and A. M. Elaiw. *Assessment and Future Directions of Nonlinear Model Predictive Control*, chapter Conditions for MPC Based Stabilization of Sampled-Data Nonlinear Systems Via Discrete-Time Approximations. Springer Berlin Heidelberg, 2007.

[66] E. Harinath, L.T. Biegler, and G. A. Dumont. Control and optimization strategies for thermo-mechanical pulping processes: Nonlinear model predictive control. *Journal of Process Control*, 21(4):519 – 528, 2011.

[67] E. N. Hartley, J. L. Jerez, A. Suardi, J. M. Maciejowski, E. C. Kerrigan, and G. A. Constantinides. Predictive control using an FPGA with application to aircraft control. *IEEE Transactions on Control Systems Technology*, 22(3):1006–1017, 2014.

[68] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[69] R. Huang, S. C. Patwardhan, and L. T. Biegler. Robust stability of nonlinear model predictive control with extended Kalman filter and target setting. *International Journal of Robust and Nonlinear Control*, 23:1240 – 1264, 2013.

[70] R. Huang, V. M. Zavala, and L. T. Biegler. Advanced step nonlinear model predictive control for air separation units. *Journal of Process Control*, 19(4):678 – 685, 2009.

[71] M. Huntley and D.G. Redalje. CO_2 mitigation and renewable oil from photosynthetic microbes: A new appraisal. *Mitigation and Adaptation Strategies for Global Change*, 12:573–608, 2007.

[72] G. A. Ifrim, M. Titica, M. Barbu, L. Boillereaux, G. Cogne, S. Caraman, and J. Legrand. Multivariable feedback linearizing control of chlamydomonas reinhardtii photoautotrophic growth process in a torus photobioreactor. *Chem. Eng. J.*, 218:191–203, 2013.

[73] A. Isidori. *Nonlinear Control Systems: An Introduction*. Communications and Control Engineering. Springer Berlin Heidelberg, 2013.

[74] A. Jadbabaie and J. Hauser. On the stability of receding horizon control with a general terminal cost. *IEEE Transactions on Automatic Control*, 50(5):674–678, 2005.

[75] J. V. Kadam and W. Marquardt. *Sensitivity-based solution updates in closed-loop dynamic optimization; 1. ed.* Elsevier, Oxford, 2005.

[76] A. Kasperski. *Discrete Optimization with Interval Data: Minmax Regret and Fuzzy Approach*. Studies in Fuzziness and Soft Computing. Springer Berlin Heidelberg, 2008.

[77] S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293.

[78] E. C. Kerrigan and J. Maciejowski. Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. *Int. J. Robust Nonlinear Control*, 14:395–413, 2004.

[79] B. Kouvaritakis and M. Cannon, editors. *Non-linear Predictive Control: Theory and Practice*. Institution of Engineering and Technology, 2001.

[80] D. Kraft. *Computational Mathematical Programming*, chapter On Converting Optimal Control Problems into Nonlinear Programming Problems, pages 261–280. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.

[81] Y. Kuriki and T. Namerikawa. Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. In *2015 European Control Conference (ECC)*, pages 3079–3084, 2015.

[82] P. L. Lee and G. R. Sullivan. Generic model control (GMC). *Comput. Chem. Eng.*, 12:6:573–580, 1988.

[83] M. A. Lelić and M. B. Zarrop. Generalized pole-placement self-tuning controller part 1, basic algorithm. *International Journal of Control*, 46(2):547–568, 1987.

[84] W. C. Li and L. T. Biegler. Process control strategies for constrained nonlinear systems. *Industrial & Engineering Chemistry Research*, 27(8):1421–1433, 1988.

[85] D. Limon, T. Alamo, and E. F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 4, pages 4619–4624, 2002.

[86] D. Limon, T. Alamo, and E.F. Camacho. Robust stability of min-max MPC controllers for nonlinear systems with bounded uncertainties. *Proceeding of the mathematical Theory of Networks and Systems*, 2004.

[87] D. Limon, T. Alamo, D. M. Raimondo, D. Muñoz Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[88] D. Limon, T. Alamo, F. Salas, and E.F. Camacho. Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties. *Automatica*, 42(5):797 – 803, 2006.

[89] D. A. Linkers and M. Mahfonf. *Advances in Model-Based Predictive Control*, chapter Generalized Predictive Control in Clinical Anesthesia. Oxford University Press, 1994.

[90] J.M. Maciejowski. *Predictive Control: With Constraints*. Prentice Hall, 2002.

[91] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer. Robust model predictive control for nonlinear discrete time systems. *International Journal of Robust and Nonlinear Control*, 13:229–246, 2003.

[92] L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42(7):1231 – 1236, 2006.

[93] L. Magni and R. Scattolini. *Assessment and Future Directions of Nonlinear Model Predictive Control*, chapter Robustness and Robust Design of MPC for Nonlinear Discrete-Time Systems, pages 239–254. Springer Berlin Heidelberg, 2007.

[94] L. Magni and R. Scattolini. *Robustness and robust design of MPC for nonlinear discrete-time systems*, volume 358. Springer-Verlag, 2007.

[95] L. Magni and R. Sepulchre. Stability margins of nonlinear receding-horizon control via inverse optimality. *Systems & Control Letters*, 32(4):241 – 245, 1997.

[96] L. Mailleret, O. Bernard, and J. P. Steyer. Nonlinear adaptive control for bioreactors with unknown kinetics. *Automatica*, 40:1379–1385, 2004.

[97] N. I. Marcos, M. Guay, and D. Dochain. Output feedback adaptive extremum seeking control of a continuous stirred tank bioreactor with monod's kinetics. *J. of Process Control*, 14:807–818, 2004.

[98] P. Masci, F. Grognard, and O. Bernard. Microalgal biomass surface productivity optimization based on a photobioreactor model. In *11th IFAC Symposium on Computer Applications in Biotechnology*, pages 180–185, 2010.

[99] D.Q. Mayne. Control of constrained dynamic systems. *European Journal of Control*, 7:87–99, 2001.

[100] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[101] F.B. Metting. Biodiversity and application of microalgae. *J. of Ind. Microbiology & Biotechnology*, 17:477–489, 1996.

[102] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[103] J. Monod. *Recherches sur la croissance des cultures bactériennes*. Actualités scientifiques et industrielles. Hermann, 1958.

[104] M. Morari and P.J. Campo. Robust model predictive control. In *American Control Conference*, pages 1021–1026, 1987.

[105] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[106] E. Mosca, J. M. Lemos, and J. Zhang. Stabilizing I/O receding horizon control. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, volume 4, pages 2518–2523, 1990.

[107] R. Munoz-Tamayo, P. Martinon, G. Bougaran, F. Mairet, and O. Bernard. Getting the most out of it: optimal experiments for parameter estimation of microalgae growth models. *J. of Process Control*, 24(6):991–1001, 2014.

[108] Z. K. Nagy and R. D. Braatz. Robust nonlinear model predictive contro of fedbatches processes. *AIChE Journal*, 49(7):1776–1786, 2009.

[109] X. M. Nguyen, F. Lawayeb, P. Rodriguez-Ayerbe, D. Dumur, and A. Mouchette. Nonlinear model predictive control of steel slab walking-beam reheating furnace based on a numerical model. In *Proc. of the 2014 IEEE MSC*, pages 191–196, 2014.

[110] J. E. Normey-Rico, J. Gómez-Ortega, and E. F. Camacho. A smith-predictor-based generalised predictive controller for mobile robot path-tracking. *Control Engineering Practice*, 7(6):729–740, 1999.

[111] M. Olaizola. Commercial development of microalgal biotechnology: from the test tube to the marketplace. *Biomolecular Eng.*, 20:459–466, 2003.

[112] S. Olaru and D. Dumur. A parameterized polyhedra approach for explicit constrained predictive control. In *43rd IEEE Conference on Decision and Control*, volume 2, pages 1580–1585 Vol.2, 2004.

[113] T. Parisini and R. Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443 – 1451, 1995.

[114] J. C. H. Peeters and P. H. C. Eilers. The relationship between light intensity and photosynthesis. *Hydrobiological Bulletin*, 12:134–136, 1978.

[115] G. Pin, D.M. Raimondo, L. Magni, and T. Parisini. Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Trans. on Auto. Cont.*, 54(7):1681–1687, 2009.

[116] L.S Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, , and E.F. Mishchenko. *The Mathematical Theory of Optimal Processes*. New York, 1964.

[117] A. I. Propoi. *Automn Remote Control*, 24, 1963.

[118] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[119] D.M. Raimondo, D. Limon, T. Alamo, and L. Magni. *Robust Model Predictive Control Algorithms for Nonlinear Systems: An Input-to-State Stability Approach*. Model Predictive Control, Tao Zheng (Ed.), 2010.

[120] D.M. Raimondo, D. Limon, M. Lazar, L. Magni, and E.F. Camachp. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 5:5–21, 2009.

[121] R. Ramine and K. M. Raman. Model algorithmic control (MAC); basic theoretical properties. *Automatica*, 18(4):401 – 414, 1982.

[122] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993.

[123] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[124] L.D. Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky. *Automotive Model Predictive Control: Models, Methods and Applications*. Springer London, 2010.

[125] J. Richalet. *Pratique de la commande prédictive*. Hermes, 1992.

[126] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251 – 1274, 1993.

[127] J. Richalet, S. Abu el Ata-Doss, C. Arber, H.B. Kuntze, A. Jacubash, and W. Schill. Predictive functional control application to fast and accurate robots. In *In Proc. 10th IFAC Congress*, 1987.

[128] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Algorithmic control of industrial processes. In *4th IFAC Symposium on Identification and System Parameter Estimation*, 1976.

[129] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Application to industrial processes. *Automatica*, 14(2):413–428, 1978.

[130] J. A. Rossiter and B. Kouvaritakis. Constrained stable generalised predictive control. *IEE Proceedings D - Control Theory and Applications*, 140(4):243–254, 1993.

[131] M. Rugabotti, D.M. Raimondo, A. Ferrara, and L. Magni. Robust model predictive control of continuous-time sampled data nonlinear systems with interval sliding mode. *IEEE Transaction on Automatic Control*, 56(3):556–570, 2010.

[132] J.M.M. Sánchez and J. Rodellar. *Adaptive Predictive Control: From the Concepts to Plant Optimization*. Prentice-Hall international series in systems and control engineering. Prentice Hall, 1996.

[133] F.L. Santamaria and J.M. Gómez. Economic oriented nmpc for an extractive distillation column using an index hybrid dae model based on fundamental principles. *Industrial & Engineering Chemistry Research*, 54(24):6344–6354, 2015.

[134] L.O. Santos, L. Dewasme, D. Coutinho, and A. Vande Wouwer. Nonlinear model predictive control of fed-batch cultures of micro-organisms exhibiting overflow metabolism: Assessment and robustness. *Computers & Chemical Engineering*, 39:143 – 151, 2012.

[135] A. H. Sayed, V. H. Nascimento, and F. A. M. Cipparrone. A regularized robust design criterion for uncertain data. *SIAM J. MAT. ANAL. APPL.*, 32:4:1120–1142, 2002.

[136] R. Scattolini. Architectures for distributed and hierarchical model predictive control-A review. *Journal of Process Control*, 19(5):723 – 731, 2009.

[137] P.O.M. Scokaert, D.Q. Mayne, and J.B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.

[138] D. Selisteanu, E. Petre, and V. Rasvan. Sliding mode and adaptive sliding mode control of a class of nonlinear bioprocesses. *Inter. J. of Adap. Contr. and Sig. Proc.*, 21:795–822, 2007.

[139] E.D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34(4):435–443, 1989.

[140] P. Spolaore, C. Joannis-Cassan, E. Duran, and A. Isambert. Commercial applications of microalgae. *J Biosci. Bioeng.*, 101:87–96, 2006.

[141] S. Tebbani, D. Dumur, G. Hafidi, and A. Vande Wouwer. Nonlinear Predictive Control of fed-batch Cultures of E. coli. *Chem. Eng. & Tech.*, 33:1112–1124, 2010.

[142] S. Tebbani, F. Lopes, and G. Becerra-Celis. Nonlinear control of continuous cultures of *Porphyridium purpureum* in a photobioreactor. *Chem. Eng. Sci.*, 123:207–219, 2015.

[143] S. Tebbani, F. Lopes, R. Filali, D. Dumur, and D. Pareau. Nonlinear predictive control for maximization of CO_2 bio-fixation by microalgae in a photobioreactor. *Bioprocess Biosyst. Eng.*, 37:83–97, 2014.

[144] S. Tebbani, M. Titica, C. Join, M. Fliess, and D. Dumur. Model-based versus model-free control designs for improving microalgae growth in a closed photobioreactor: Some preliminary comparisons. In *22nd MED Conference*, pages –, 2016.

[145] M. K. Toroghi, G. Goffaux, and M. Perrier. Observer based backstepping controller for microalgae cultivation. *Ind. & Eng. Chem. Res.*, 52:7482–7491, 2013.

[146] V. Utkin and J. Shi. Integral sliding mode in systems operating under uncertainty conditions. *Proc. of the 35th Conference on Decision and Control*, 4:4591–4596, 1996.

[147] V.S. Vassiliadis. *Computational solution of dynamic optimization problems with general differential algebric constraints*. PhD thesis, University of London, 1993.

[148] Z. Y. Wan and M. V. Kothare. Efficient scheduled stabilizing output feedback model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 49(10):1172–1177, 2004.

[149] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB®*. Advances in Industrial Control. Springer, 2009.

[150] R. Wang, G. P. Liu, W. Wang, D. Rees, and Y. B. Zhao. h_∞ control for networked predictive control systems based on the switched lyapunov function method. *IEEE Transactions on Industrial Electronics*, 57(10):3565–3571, 2010.

[151] R. Wang, B. Wang, G. P. Liu, W. Wang, and D. Rees. h_∞ controller design for networked predictive control systems based on the average dwell-time approach. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(4):310–314, 2010.

[152] L. W**ü**rth, R. Hannemann, and W Marquardt. Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization. *Journal of Process Control*, 19(8):1277 – 1288, 2009.

[153] B.E. Ydstie. Extended horizon adaptive control. In *In Proc. 9th IFAC World Congress*, 1984.

[154] L.A. Zadeh and B.H. Whalen. On optimal contro and linear programming. *IRE Trans. Aut. Control*, 7(4):729 – 740, 1962.

[155] J. Zarate Florez, J.J. Martinez, G. Besancon, and D. Faille. Explicit coordination for mpc-based distributed control with application to hydropower valleys. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 830–835, 2011.

[156] V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86 – 93, 2009.

[157] A. Zheng and M. Morari. Stability of model predictive control with soft constraints. *Internal Report. California Institute of Technology*, 7(4):729 – 740, 1994.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Robustification de la commande prédictive non linéaire - Application à des procédés pour le développement durable

Mots clés : Commande prédictive non linéaire, Robustesse, Systèmes incertains, Bioprocédé

Résumé : Les dernières années ont permis des développements très rapides, tant au niveau de l'élaboration que de l'application, d'algorithmes de commande prédictive non linéaire (CPNL), avec une gamme relativement large de réalisations industrielles. Un des obstacles les plus significatifs rencontré lors du développement de cette commande est lié aux incertitudes sur le modèle du système. Dans ce contexte, l'objectif principal de cette thèse est la conception de lois de commande prédictives non linéaires robustes vis-à-vis des incertitudes sur le modèle. Classiquement, cette synthèse peut s'obtenir via la résolution d'un problème d'optimisation min-max. L'idée est alors de minimiser l'erreur de suivi de la trajectoire optimale pour la pire réalisation d'incertitudes possible. Cependant, cette formulation de la commande prédictive robuste induit une complexité qui peut être élevée ainsi qu'une charge de calcul importante, notamment dans le cas de systèmes multivariables, avec un nombre de paramètres incertains élevé. Pour y remédier, la principale approche proposée dans ces travaux consiste à simplifier le problème d'optimisation min-max, via l'analyse de sensibilité du modèle vis-à-vis de ses paramètres afin d'en réduire le temps de calcul. Dans un premier temps, le critère est linéarisé autour des valeurs nominales des paramètres du modèle. Les variables d'optimisation sont soit les commandes du système soit l'incrément de commande sur l'horizon temporel. Le problème d'optimisation initial est alors transformé soit en un problème convexe, soit en un problème de minimisation unidimensionnel, en fonction des contraintes imposées sur les états et les commandes. Une analyse de la stabilité du système en boucle fermée est également proposée. En dernier lieu, une structure de commande hiérarchisée combinant la commande prédictive robuste linéarisée et une commande par mode glissant intégral est développée afin d'éliminer toute erreur statique en suivi de trajectoire de référence. L'ensemble des stratégies proposées est appliqué à deux cas d'études de commande de bioréacteurs de culture de microorganismes.

Title : Robustification of Nonlinear Model Predictive Control - Application to sustainable development processes

Keywords : Nonlinear model predictive control, Robustness, Uncertain Systems, Bioprocess

Abstract : The last few years have led to very rapid developments, both in the formulation and the application of Nonlinear Model Predictive Control (NMPC) algorithms, with a relatively wide range of industrial achievements. One of the most significant challenges encountered during the development of this control law is due to uncertainties in the model of the system. In this context, the thesis addresses the design of NMPC control laws robust towards model uncertainties. Usually, the above design can be achieved through solving a min-max optimization problem. In this case, the idea is to minimize the tracking error for the worst possible uncertainty realization. However, this robust approach tends to become too complex to be solved numerically online, especially in the case of multivariable systems with a large number of uncertain parameters. To address this shortfall, the main proposed approach consists in simplifying the min-max optimization problem through a sensitivity analysis of the model with respect to its parameters, in order to reduce the calculation time. First, the criterion is linearized around the model parameters nominal values. The optimization variables are either the system control inputs or the control increments over the prediction horizon. The initial optimization problem is then converted either into a convex optimization problem, or a one-dimensional minimization problem, depending on the nature of the constraints on the states and commands. The stability analysis of the closed-loop system is also addressed. Finally, a hierarchical control strategy is developed, that combines a robust model predictive control law with an integral sliding mode controller, in order to cancel any tracking error. The proposed approaches are applied through two case studies to the control of microorganisms culture in bioreactors.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France