ENHANCING CLOSED-LOOP CONTROL PERFORMANCE WITH MODEL-BASED CONTROLLERS: ALGORITHMS AND APPLICATIONS FOR TIME-DELAYED SYSTEMS

Sergio Andres Castaño Giraldo

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Química.

Orientadores: Argimiro Resende Secchi
Príamo Albuquerque Melo Junior

Rio de Janeiro
Setembro de 2023

ENHANCING CLOSED-LOOP CONTROL PERFORMANCE WITH MODEL-BASED CONTROLLERS: ALGORITHMS AND APPLICATIONS FOR TIME-DELAYED SYSTEMS

Sergio Andres Castaño Giraldo

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Orientadores: Argimiro Resende Secchi
Príamo Albuquerque Melo Junior

Aprovada por: Prof. Argimiro Resende Secchi
Prof. Príamo Albuquerque Melo Junior
Prof. Julio Elias Normey Rico
Prof. Márcio André Fernandes Martins
Prof. Mauricio Bezerra de Souza Junior

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2023

Giraldo, Sergio Andres Castaño

Enhancing Closed-Loop Control Performance with Model-Based Controllers: Algorithms and Applications for Time-Delayed Systems/Sergio Andres Castaño Giraldo. – Rio de Janeiro: UFRJ/COPPE, 2023.

XXIII, 155 p.: il.; 29, 7cm.

Orientadores: Argimiro Resende Secchi

Príamo Albuquerque Melo Junior

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Química, 2023.

Referências Bibliográficas: p. 132 – 140.

1. Model Predictive Control (MPC). 2. Filtered Smith Predictor (FSP). 3. Model-Plant Mismatch (MPM). 4. Unmeasured Disturbance (UD). 5. Performance Diagnosis. 6. Tuning. I. Secchi, Argimiro Resende *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

"We are just waves in time and space, changing continuously, and the illusion of individuality is produced through the concatenation of the rapidly succeeding phases of existence. What we define as likeness is merely the result of the symmetrical arrangement of molecules which compose our body."

"If you want to find the secrets of the universe, think in terms of energy, frequency and vibration."

Nikola Tesla

Agradecimentos

Primeiramente, expresso minha sincera gratidão à vida, essa força infinita que me trouxe até aqui, me presenteando com oportunidades e desafios que jamais ousei sonhar. Nunca imaginei que poderia trilhar o caminho acadêmico até o doutorado, contudo, a vida, com sua magia imprevisível, mudou o curso de minha história.

Quando a Colômbia, minha terra natal, se transformou em um pontinho no horizonte e o Brasil se tornou minha nova morada procurando ser professor algum dia, uma porta se abriu. Atravessei essa porta e descobri um novo universo de possibilidades, o universo da pesquisa.

Nesse novo universo, um desafio colossal se apresentou. Ingressar no doutorado em Engenharia Química, vindo de uma formação em Engenharia de Controle e Automação, foi como adentrar um labirinto de significados desconhecidos. Os jargões da área, os novos estudos de caso, as diversas disciplinas - tudo isso era como um terreno desconhecido que eu precisava desbravar. Houve noites de insônia, dias de luta, momentos em que o desânimo tentou me derrubar.

Mas eu resisti. Descobri em mim um espírito resiliente, uma vontade de superação que me fez persistir, mesmo diante dos obstáculos mais desafiadores. A cada passo, a cada palavra desconhecida que se tornava familiar, eu crescia um pouco mais. Fui tecendo a teia do conhecimento, não com a genialidade de muitos dos meus colegas e amigos, mas com a determinação de quem está apaixonado pelo que faz.

Ao longo dessa estrada repleta de desafios, o elo já conhecido do controle de processos, que havia sido revelado em minha jornada anterior no mestrado, se tornou ainda mais forte na minha incursão na Engenharia Química. Esta familiaridade, juntamente com o desejo de explorar novos territórios, foi o que motivou a minha escolha por este campo.

Em segundo lugar, desejo estender meu sincero agradecimento, profunda admiração e renovada inspiração aos meus orientadores, Professor Argimiro e Professor Príamo. Sob o guarda-chuva de suas orientações, tive a oportunidade de expandir meus horizontes, ampliar meu conhecimento, abrir minha mente para territórios não explorados além do controle. A cada um deles, agradeço pela oportunidade concedida, pelas orientações precisas e pelos ensinamentos profundos que contribuíram

imensamente para minha formação.

Ao professor Argimiro, minha gratidão se estende além das paredes da academia. Além de orientador e chefe, sempre o considerei como um amigo que, com seu farol de conhecimento e experiência, iluminou o meu caminho durante a minha estadia no Rio de Janeiro. Seu fervor e paixão pelo que faz, manifestados na sua entrega incansável - respondendo e-mails às três da manhã, estando sempre presente para fornecer orientação - foram um exemplo constante para mim. Foram além de nossas discussões acadêmicas na área de Engenharia de Sistemas em Processos, compartilhamos momentos onde o debate se estendia à vida, à política, ao trabalho, às trilhas, à música e às viagens. Essas experiências enriqueceram não só a nossa relação, mas também ampliaram minha compreensão do mundo, permitindo-me aprender não só como um engenheiro, mas também como um ser humano em constante evolução.

Ao professor Príamo, minhas palavras de gratidão transbordam a mera relação orientador-orientando. Além das discussões valiosas na área de modelagem de processos e de sua didática incomparável, que transmite um amor genuíno por cada sala de aula e pelo conhecimento em si, o professor tornou-se um amigo e um guia, auxiliando em minha formação não apenas como um profissional, mas também como uma pessoa, mostrando-me que as múltiplas facetas da vida, como a arte e a ciência, podem se entrelaçar e se complementar de maneiras inesperadas, potencializando a forma como percebemos e nos engajamos com o mundo. Isso não só enriqueceu a minha jornada de doutorado, mas também redefiniu a minha compreensão da vida e do aprendizado, incentivando-me a buscar conexões mais profundas e mais significativas em todas as áreas da minha vida..

Para minha querida Jessica, minha companheira de vida, amiga fiel e amor eterno. A ti, ofereço minhas mais profundas expressões de gratidão. Desde o início desta jornada, você esteve ao meu lado, sendo meu farol na escuridão e minha âncora em meio às tempestades. Você me apoiou, incentivou e motivou, mesmo nos momentos de a dúvida e incerteza. Em ti, encontrei a paciência e a tolerância que sustentaram minha persistência e a ternura que confortou meu coração nos momentos de desafio. Cada lugar novo, cada sorriso compartilhado, cada por do sol admirado juntos, gravou uma lembrança preciosa em nossos corações. Em meio a toda a agitação do doutorado, esses momentos de conexão com você foram as jóias que adornaram meus dias. Com você descobri que a verdadeira fortaleza não vem da superação de dificuldades sozinho, mas da habilidade de enfrentar desafios juntos, de sermos um para o outro, o abraço seguro na tormenta, a alegria no fim de um dia longo e difícil.

Para Carol, Leonardo, Rafael e Nevardo, minhas mais profundas expressões de gratidão. Como irmãos nesta jornada no Rio de Janeiro, nossa amizade se fortaleceu em meio aos desafios, aos sorrisos, às lágrimas e às risadas. Ao grupo que formamos

sob o lema "Nunca Acaba", provamos que o doutorado pode, de fato, terminar. Mas a nossa união, essa sim, persiste inabalável e infinita. Carol, minha grande amiga, sua lealdade, senso de humor e capacidade de aconselhar e apoiar nos momentos necessários foram fundamentais para mim. Você é o pilar emocional de nosso grupo, trazendo equilíbrio e alegria para nossas vidas. Nevardo, mesmo sendo o membro mais teimoso entre nós, você é um verdadeiro amigo. Seus conselhos, embora por vezes contundentes, sempre foram valiosos. Sua personalidade única trouxe uma rica dimensão à nossa amizade, um contraponto essencial ao dinamismo do grupo. Rafa, sua dança de rato inconfundível e sua habilidade em contar histórias sempre nos deixaram surpresos. Mas foi seu talento para decifrar códigos e nos guiar nas disciplinas mais árduas que realmente me fez admirar sua inteligência e amizade. E por último, mas não menos importante, Leonardo. Juntos, partilhamos incontáveis experiências, desde longas conversas até simples momentos de descontração. Tua amizade, firme como o cuscuz que nunca falte. A vocês quatro, meus queridos amigos, meus irmãos nesta jornada, minha gratidão infinita. Com vocês, a jornada do doutorado não foi apenas suportável, mas alegre e inesquecível. Vocês transformaram essa fase da minha vida em uma experiência rica e cheia de momentos de pura alegria. Obrigado por tudo.

Estendo meu sincero agradecimento aos colegas na G130 e no Lades. Especialmente para Laura, cuja amizade me proporcionou encontros ricos que oscilavam entre debates científicos e discussões lúdicas sobre filmes e videogames. A Ana Carolina, Charles, Pedro e Daniel, agradeço pelas reflexões instigantes sobre controle que transformaram o aprendizado em uma experiência de descoberta coletiva. Com Christian e Leonardo Voltolini, tive momentos preciosos de diálogos sobre a vida, que trouxeram um toque de humanidade ao rigor da academia. E a Roymel, expresso gratidão pelas conversas instrutivas sobre simulação que aprofundaram minha compreensão e apreciação desta disciplina. Cada um de vocês contribuiu de maneira única para esta etapa da minha jornada.

Os laços que construí com meus companheiros colombianos no Rio de Janeiro - Reinaldo, Nayher e César - são inestimáveis. Juntos, cruzamos muitas jornadas, compartilhando histórias, risadas, bebidas e aventuras. Apreciar as histórias de cada um de vocês, entender suas aspirações, sonhos e angústias, compartilhar nossa cultura comum enquanto explorávamos as novas experiências no Brasil, tudo isso teve um significado imenso para mim. Além do apoio constante e do diálogo enriquecedor em nossas pesquisas, compartilhamos momentos de irmandade e solidariedade em nosso caminho compartilhado como estrangeiros em busca de ampliar nossos horizontes de conhecimento - uma busca às vezes difícil de realizar na Colômbia. Sem vocês, a caminhada teria sido muito mais árdua. A todos vocês, minha gratidão profunda.

Agradeço a todos aqueles que, mesmo à distância, aguardam por mim na Colômbia, sempre oferecendo apoio inabalável e amor incondicional. Em primeiro lugar, a minha mãe, Gabriela Giraldo. Você é o farol da minha vida, o exemplo supremo de resiliência e força. Diante de adversidades e dificuldades, você encontrou maneiras de triunfar, sempre priorizando o bem-estar de seus filhos. Você nos ensinou o valor de nunca nos rendermos, de lutar por nossos sonhos. O amor, o carinho, a orientação e a compreensão que você infundiu em nós são inestimáveis. O seu espírito incansável e o seu amor abnegado são a fonte da minha inspiração e força.

Devo expressar minha gratidão à minha família, especialmente à minha avó Berta, cujo carinho e histórias sempre aquecem meu coração; e ao meu primo Julian, que sempre considero como um irmão. Ao meu irmão Diego, mesmo com nossas diferenças, é sempre uma alegria revê-lo, e ao meu pai Freddy, com suas histórias.

Um agradecimento especial também aos meus amigos na Colômbia. Richard, um amigo para a vida toda, nossas conversas filosóficas sempre me transportam de volta para casa. Carlos (Mellito), a amizade e os momentos que compartilhamos são tesouros preciosos, e a forma como sempre me acolhe é inestimável. Os amigos da faculdade - Johnny (Puma), Andres (Mono), Armando - que continuam me dando força e apoio. E, em particular, Deinis e Dayron, com quem mantenho uma constante conexão à distância. Nós compartilhamos momentos, celebramos conquistas uns dos outros e nos incentivamos a ser melhores pessoas todos os dias. Que a vida nos permita continuar a fortalecer essa bela amizade.

Sou extremamente grato ao programa de Engenharia Química da COPPE e a todos os seus funcionários, professores e comunidade acadêmica em geral. A oportunidade de estar neste programa não só contribuiu para a minha formação profissional, mas também me moldou pessoalmente, ajudando-me a evoluir como indivíduo.

Também gostaria de expressar minha profunda gratidão à CAPES, CNPq e aos projetos da Petrobras pelo suporte financeiro provido durante o curso do meu doutorado. Sem este apoio, minha jornada não teria sido possível. A generosidade e o compromisso destas instituições com o avanço da pesquisa e da educação são verdadeiramente louváveis e merecem reconhecimento.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

MELHORIA DO DESEMPENHO DE CONTROLE EM MALHA FECHADA COM CONTROLADORES BASEADOS EM MODELO: ALGORITMOS E APLICAÇÕES PARA SISTEMAS COM ATRASOS DE TEMPO

Sergio Andres Castaño Giraldo

Setembro/2023

Orientadores: Argimiro Resende Secchi
Príamo Albuquerque Melo Junior

Programa: Engenharia Química

Este trabalho de doutorado desenvolve e valida duas metodologias distintas para aprimorar o desempenho de controladores preditivos baseados em modelos, focando principalmente nas estruturas de Controle Preditivo baseado em Modelo (MPC) e nos compensadores de tempo morto (DTC). A primeira parte da pesquisa apresenta um algoritmo de sintonia inovador para controladores MPC que é híbrido, combinando dois métodos de otimização para determinar os melhores parâmetros de sintonia. Este algoritmo foi projetado para ser versátil, aplicável a uma ampla gama de sistemas, incluindo sistemas lineares e não lineares, e com custo computacional relativamente baixo. A segunda parte da pesquisa se concentra em algoritmos de diagnóstico em tempo real para sistemas monovariáveis, especificamente aplicados aos compensadores de tempo morto (DTC) para o Preditor de Smith Filtrado (FSP). Este algoritmo multifuncional não apenas detecta erros de modelo e perturbações não medidas, mas também ajusta com precisão o filtro de robustez, contribuindo para a estabilidade do sistema. Ambas as metodologias foram rigorosamente validadas através de estudos de caso, tanto simulados quanto em cenários do mundo real, destacando sua eficácia prática em melhorar o desempenho e a robustez dos sistemas de controle em malha fechada baseados numa estrutura de predição com modelo interno, trazendo implicações promissoras para a engenharia de sistemas de controle.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

ENHANCING CLOSED-LOOP CONTROL PERFORMANCE WITH MODEL-BASED CONTROLLERS: ALGORITHMS AND APPLICATIONS FOR TIME-DELAYED SYSTEMS

Sergio Andres Castaño Giraldo

September/2023

Advisors: Argimiro Resende Secchi
Príamo Albuquerque Melo Junior

Department: Chemical Engineering

This doctoral research develops and validates two distinct methodologies to enhance the performance of model-based controllers, focusing primarily on Model Predictive Control (MPC) structures and Dead Time Compensators (DTC). The first part of the research presents an innovative tuning algorithm for MPC controllers through a hybrid combination of two optimization algorithms to determine the best tuning parameters. This algorithm is designed to be versatile, applicable to a wide range of systems, including linear and non-linear ones, and with relatively low computational cost. The second part of the research focuses on real-time diagnostic algorithms for single-input, single-output systems, specifically applied to Dead Time Compensators for Filtered Smith Predictors (FSP). This multifunctional algorithm not only detects model errors and unmeasured disturbances but also accurately adjusts the robustness filter, contributing to system stability. Both methodologies have been rigorously validated through case studies, both simulated and in real-world scenarios, highlighting their practical efficacy in improving the performance and robustness of closed-loop control systems based on an internal model predictive structure, bringing promising implications for control systems engineering.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{E}$	mathematical expectation, p. 10
e_p	predicted error, p. 32
n_1	output disturbance, p. 32
$A(z^{-1})$	polynomial associated with the poles of the plant, p. 12
$B(z^{-1})$	polynomial associated with the zeros of the plant, p. 12
C	primary SISO controller of the FSP, p. 32
$C(z^{-1})$	polynomial associated with the zeros of the disturbance, p. 12
F_i^*	optimum value of the objective goal (utopia solution), p. 24
G_n	SISO model without transport delay., p. 32
L_n	transport delay of the nominal SISO model., p. 32
M_1	order of polynomial $A(z^{-1})$ of the ARX model, p. 44
M_2	order of polynomial $B(z^{-1})$ of the ARX model, p. 44
$Mk(\kappa)$	maximal value of the function $K(s)$, p. 26
$Ms(\kappa)$	maximal sensitivity value, p. 26
N_s	collected output samples, p. 9
P	real SISO plant., p. 32
P_n	nominal model of the SISO process., p. 32
Δ	difference operator, p. 18
Δu	incremental control action, p. 18
Γ	Robust performance function, p. 26

Γ_{REF}	minimal possible value of the function Γ, p. 26
$\bar{\sigma}(\mathbf{M})$	maximal singular value of the transfer function matrix $\mathbf{M}$, p. 26
$\beta(\mathbf{G})$	the condition number of the matrix $\mathbf{G}$, p. 27
$\mathbf{L}(z)$	the nominal MIMO delay model, p. 39
ϵ	slack variable of the cost function., p. 19
η_{MV}	Harris index, p. 15
κ	square subsystems originated from the full model, p. 25
$\mathbf{A}(z^{-1})$	a diagonal polynomial matrix with elements $A_i(z^{-1})$ equal to the least common multiple of the denominators $A_{ij}(z^{-1})$, p. 37
$\mathbf{B}(z^{-1})$	a polynomial matrix with elements $B_{ij}(z^{-1})$, p. 38
$\mathbf{D}(z^{-1})$	a polynomial diagonal matrix with elements z^{-d_i}, p. 37
$\mathbf{F_r}(z)$	the MIMO predictor filter of the FSP, p. 39
$\mathbf{G_n}(z)$	fast model of the FSP, p. 39
$\mathbf{G}_{CL}^{ref}(z^{-1})$	desired MPC closed-loop transfer function, p. 29
$\mathbf{L_B}$	lower bounds of the decision variables, p. 24
$\mathbf{P_n}(z)$	the nominal process model of the FSP, p. 39
$\mathbf{Q}$	positive semi-definite diagonal matrix used to weight the squared error between the predicted system output and the desired reference or setpoint of the MPC function., p. 19
$\mathbf{R}_A(z^{-1})$	denominator of $\mathbf{G}_{CL}^{ref}(z^{-1})$, p. 29
$\mathbf{R}_B(z^{-1})$	numerator of $\mathbf{G}_{CL}^{ref}(z^{-1})$, p. 29
$\mathbf{S_t}$	diagonal weighting matrix of LTT, p. 24
$\mathbf{T}(z^{-1})$	monic polynomial matrix representing the colouring polynomials of the noise, p. 38
$\mathbf{U_B}$	upper bounds of the decision variables, p. 24

$\mathbf{W}$ positive semi-definite diagonal matrix used to weight the squared change in the control input of the MPC function., p. 19

$\dot{\mathbf{x}}$ process model (state-space representation), p. 63

$\dot{\mathbf{x}}^R$ state-space model for the reference trajectory in the VNS algorithm, p. 63

$\mathbf{e^d}(k)$ the stochastic disturbance error, p. 44

$\mathbf{e^p}(k)$ output error model, p. 45

$\mathbf{e}(k)$ the $n \times 1$ noise vector, p. 38

$\mathbf{f_r}$ free response of the MPC, p. 39

$\mathbf{r}$ reference vector., p. 19

$\mathbf{r}_s$ reference of the internal model when the VNS algorithm is used, p. 58

$\mathbf{u}$ manipulated variables vector., p. 19

$\mathbf{x_{dv}}$ vector of decision variables of MPC self-tuning methods, p. 23

$\mathbf{x}$ state variables vector., p. 19

$\mathbf{y}$ controlled variables vector., p. 19

ρ penalty associated with violating soft constraints., p. 19

ρ_g the move suppression weight of LQG, p. 10

$\sigma^2_{\Delta u}$ process input variance, p. 42

σ^2_y process output variance, p. 9

σ^2_{MV} the output variance of minimum variance controller, p. 15

$\mathbf{P}(s)$ the plant with multiple time delays, p. 39

θ the magnitude of model uncertainty of the min-max optimization problem, p. 27

θ_t the tuning horizon, p. 23

d discrete dead-time, p. 12

d_n	nominal discrete time delay of the FSP., p. 34
$g_{ij}(z^{-1})$	the SISO transfer functions without the common delay, p. 38
k	discrete time (multiple of sampling time), p. 9
m_j	control horizon value of j-th plant input., p. 19
n_u	number of manipulated variables., p. 19
n_y	number of controlled variables., p. 19
n_{ai}	order of the polynomials $A_i(z^{-1})$, p. 38
n_{bi}	order of the polynomials $B_i(z^{-1})$, p. 38
p	prediction horizon value., p. 19
q	load disturbance, p. 32
$r(k+j)$	future set-point, p. 18
$rRPN(\kappa)$	relative robust performance number of the scenario κ, p. 25
$u(k)$	control action, p. 11
$v(k)$	stochastic perturbation, p. 12
w'	the current tuning step, p. 24
$y(k)$	process output, p. 9
$y_i^{ref}(k)$	discretized reference trajectory of output i, p. 23
ω_i	relative weight for the i-th objective function of GAM algorithm, p. 53
$\mathrm{Q}_0(z^{-1})$	nominal disturbance model, p. 44
n_2	measured noise, p. 32
rt	desirable rise time, p. 25
$\mathrm{S}_d(z)$	designed sensitivity function, p. 47
$\mathrm{S}_o(z)$	nominal output sensitivity transfer function, p. 49

List of Abbreviations

ARMAX	autoregressive–moving-average with exogenous inputs model, p. 12
ARMA	autoregressive–moving-average model, p. 16
CARIMA	controlled auto-regressive integrated moving average, p. 38, 42
CTT	compromise tuning technique, p. 23
DTC-GPC	dead-time compensator generalized predictive controller, p. 37
DTC	dead-time compensator, p. 32
FSP	Filtered Smith Predictor, p. 33
GAM	goal attainment method, p. 52
GPC	generalized predictive control, p. 11
HOF	the Shell Heavy Oil Fractionator, p. 69
LQG	linear quadratic Gaussian, p. 2
LTT	lexicographic tuning technique, p. 23
MIMO	multi-input multi-output, p. 2
MPCT	model predictive control tuning approach, p. 55
MPC	model predictive control, p. 2
MPM	model-plant mismatch, p. 98
MVC	minimal variance control, p. 2
PID	proportional-integral-differential, p. 2
PSO	particle swarm optimization, p. 28
SISO	single-input single-output, p. 2

SP	Smith predictor, p. 32
UD	unmeasured disturbance, p. 98
VNS	variable neighborhood search, p. 52

Chapter 1

Introduction

1.1 Relevance and Motivation

The overarching goal of control theory is developing methods to drive a system's behavior as closely as possible to a desired performance specification. Among the array of control techniques, model-based predictive controllers have gained considerable attention due to their ability to anticipate future events and, thus, provide an optimized control strategy. Central to these controllers, which encompass Model Predictive Control (MPC) and Dead-Time Compensators (DTC), is the notion of predicting the future behavior of a system based on an internal model and employing that prediction to make informed control decisions (CAMACHO and BORDONS, 2002).

However, as promising as this predictive approach is, it is not devoid of challenges. One key challenge lies in the tuning of control parameters. In complex, multivariable systems, the number of parameters grows proportionally with the system size, complicating their precise tuning (BAGHERI and KHAKI-SEDIGH, 2014). Moreover, selecting prediction and control horizons and adjusting weight matrices can heavily influence the system's closed-loop performance (YAMASHITA *et al.*, 2016). Tuning these parameters is a procedure that has proven particularly challenging, with no definitive methodology universally recognized as the most effective solution (GARRIGA and SOROUSH, 2010).

Another fundamental difficulty arises from the inherent characteristic of model-based controllers: they rely on a model of the system to predict future behavior. Although such predictive ability is their strength, it is also their Achilles' heel. Over time, the model's ability to accurately predict the system's behavior can degrade due to unmeasured disturbances, changes in system dynamics, or process nonlinearity. When a system deviates from the controller's model, control performance can deteriorate, leading to a less than optimal response (SHRIDHAR and COOPER,

1997).

In this context, process models play an essential role in any control structure design because (i) classical control schemes use models offline in combination with tuning guidelines and (ii) model-based controllers use the model online to generate output predictions and take corrective actions (YERRAMILLI and TANGIRALA, 2016). Therefore, the quality of the model significantly affects the closed-loop performance. This performance is critical to industrial processes' product quality and system safety. In the last decades, the monitoring and performance assessment of closed-loop control has gained critical attention, giving rise to several techniques (LING *et al.*, 2017). Some of these are based on the concept of minimal variance control (MVC), initially developed by ÅSTRÖM (1970). Then, it was produced by HARRIS (1989), who evaluated the best achievable benchmark from normal closed-loop data. It has subsequently been extended to multi-input multi-output (MIMO) systems in HARRIS *et al.* (1996) and HARRIS *et al.* (1999), feedforward/feedback control in DESBOROUGH and HARRIS (1993), proportional-integral-differential (PID) control in FU *et al.* (2012), Run-to-Run (RtR) control in CHEN *et al.* (2009) and model predictive control (MPC) system in LEE *et al.* (2008). As an alternative to the MVC benchmark, in POUR *et al.* (2010) it was proposed the optimal linear quadratic Gaussian (LQG), which provides a trade-off curve that displays the minimal achievable variance of the input and output of the system. Nevertheless, some authors disagree on using LQG/MVC for control structure assessment, mainly in model-based controllers like an MPC, as they deem it an unattainable model for most real-life applications (BOTELHO *et al.*, 2015).

The challenges in tuning and modeling are not the only hurdles for optimizing control systems. Even a well-tuned and modeled system can begin to falter if not properly maintained. Maintenance plays a crucial role in preserving the efficiency and efficacy of the control system over time. As illustrated in Figure 1.1, the lack of adequate maintenance can lead to a gradual deterioration of the control system's performance, undermining the optimizations that have been achieved (CAMPOS *et al.*, 2013).

This work is focused on two key areas of model-based predictive controllers: the tuning methods for the MPC and the model degradation detection in DTC, specifically within the Smith predictor framework for Single-Input Single-Output (SISO) systems.

Firstly, the research explores the tuning methods for the MPC, aiming to develop a universal strategy that can be applied regardless of the control scheme (be it linear or non-linear) and the system type (whether square or non-square). By working towards a more versatile and robust tuning procedure, the proposed research aims to enhance the application of MPC across various system architectures

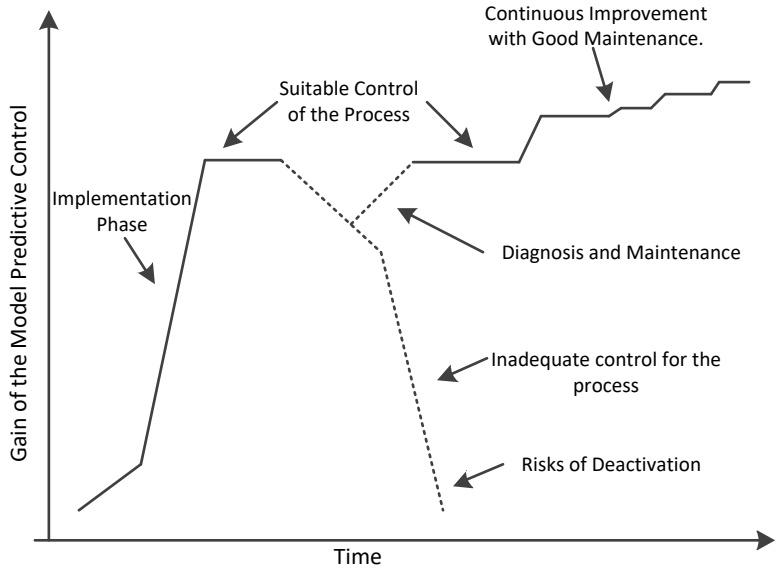


Figure 1.1: Control system's life cycle.

and control strategies. This can potentially reduce the complexities associated with manual tuning and open the door to more automated, adaptive approaches (QIN and BADGWELL, 2003).

Subsequently, the research delves into the domain of DTCs, explicitly focusing on the filtered Smith predictor for SISO systems. This part of the study aims to diagnose the internal model's predictive capability and devise methods to detect unmeasured disturbances. The ultimate goal is to improve the robustness of the control loop by developing an auto-tuning mechanism for both the primary controller and the robustness filter in the control structure. By assessing the degree of model degradation and detecting the onset of unmeasured disturbances, it is expected to devise strategies to maintain control performance, thus enhancing the utility of the Smith predictor in DTCs.

These two approaches address the fundamental challenges in model-based predictive controllers and target improvements that can significantly enhance closed-loop control performance in industrial applications.

1.2 Objectives

1.2.1 General Objective

The central objective of this thesis is to develop and validate methodologies for enhancing the performance of model-based controllers, specifically, Model Predictive

Control (MPC) and Dead Time Compensator (DTC). The research aims to address the issues arising from dominant time delays, tuning parameters, the predictive capacity of the internal model, and unmeasured disturbances in industrial process control systems.

1.2.2 Specific Objectives

- To construct an algorithm that effectively identifies optimal MPC tuning parameters using a hybrid combination of two optimization algorithms. The algorithm is expected to be applicable to various system types, including linear, non-linear, square, and non-square systems.
- To develop a real-time diagnostic algorithm to monitor the performance of the control loop in DTC systems. The algorithm should be capable of identifying the degradation of the internal model or the presence of unmeasured disturbances.
- To explore the effectiveness of the proposed algorithms in enhancing the performance of the Filtered Smith Predictor (FSP) in single-variable systems, specifically addressing the issues associated with dead-time compensators (DTCs).
- To demonstrate the application of the proposed methodologies and algorithms through case studies, thus highlighting their practical applicability and effectiveness in enhancing industrial process control.

1.3 Document Arrangement

This doctoral thesis is organized into five chapters, which are structured as follows:

- **Chapter 1: Introduction** - This chapter lays the foundation for the thesis by introducing the research's relevance and motivation. The general and specific objectives of the study are clearly stated. The document arrangement and a list of publications stemming from this research are presented.
- **Chapter 2: Literature Review** - A comprehensive review of the literature on predictive controllers, principles of model-based control, challenges and improvements in the tuning of predictive controllers, and control of processes with dead time is provided in this chapter. The chapter also discusses the detection and diagnosis of predictive controller issues and concludes with some final remarks.

- **Chapter 3: Tuning of Model Predictive Controllers Based on Hybrid Optimization** - This chapter focuses on the proposed methodology for tuning model predictive controllers using hybrid optimization techniques. It discusses in detail the Model Predictive Control Tuning Approach (MPCT) and presents various simulation case studies to validate the proposed methodology.
- **Chapter 4: Filtered Smith Predictor Monitoring, Diagnosis, and Self-tuning due to Unmeasured Abrupt Load Disturbance or Model Plant Mismatch** - This chapter details another significant part of the research which involves monitoring, diagnosis, and self-tuning of the Filtered Smith Predictor. It presents a detailed methodology, case studies, and experimental application of the proposed approach.
- **Chapter 5: Conclusions** - This final chapter summarizes the key findings of the research and discusses the implications of the research in the control systems engineering field. It offers a recap of the main contributions of the study and suggests future research directions based on the results.

In addition to these chapters, this thesis includes an extensive list of references and several appendices. The appendices provide additional material related to the adequacy of the process model to be controlled, DTC-GPC implementation example, comparative performance evaluation of PI control with and without model plant mismatch (MPM) and unmeasured disturbance (UD) monitoring structure, and PI controller performance on the Temperature Control Lab (TCLab) system. There is also an appendix that shows the construction circuit of the TCLab which is a board oriented for the teaching and study of control theory, along with its implementation code. These supplementary materials offer a more in-depth exploration of the research's methodologies and findings.

1.4 Publications

This section presents an overview of the research outputs produced during this doctoral research, published in peer-reviewed journals, and presented at academic conferences. These works provide the basis for the methodologies and algorithms presented in this thesis, demonstrating their development, validation, and application.

1.4.1 Publications in Index Journals

- Giraldo, S.A.C., Melo, P.A., & Secchi, A.R. (2022). Tuning of Model Predictive Controllers Based on Hybrid Optimization. *Processes*,

10(2), 351. https://doi.org/10.3390/pr10020351

- Giraldo, S.A.C., Melo, P.A., & Secchi, A.R. (2023). Filtered Smith Predictor Monitoring, Diagnosis and Self-tuning due to Unmeasured Abrupt Load Disturbance or Model Plant Mismatch. *Journal of Process Control* (Under Evaluation).

1.4.2 Conference Papers

- Giraldo, S.A.C., Melo, P.A., & Secchi, A.R. (2019). Tuning of Model Predictive Control Based on Hybrid Optimization. *IFAC-PapersOnLine*, 52(1), 136-141. https://doi.org/10.1016/j.ifacol.2019.06.050

- Garcia, C.A., Giraldo, S.A.C., & Secchi, A.R. (2023). Sintonia otimizada do LMPC para uma coluna de destilação reativa em escala de planta piloto. To be presented at the 24th Brazilian Congress of Chemical Engineering in October 2023 (COBEQ 2023).

- Giraldo, S.A.C., Melo, P.A., & Secchi, A.R. (2023). Método de ajuste de MPC para sistemas não quadrados baseado em otimização híbrida. To be presented at the 24th Brazilian Congress of Chemical Engineering in October 2023 (COBEQ 2023).

1.4.3 Additional Research Publications

In addition to the research directly linked to this thesis, the doctoral journey has led to a series of publications focusing on various applications of Model Predictive Control (MPC). These works, although not directly connected to the main focus of this thesis, have significantly enriched the research process. They have provided diverse perspectives, deepened the understanding of MPC applications, and offered additional avenues of research exploration. These publications offer a detailed account of the academic contributions achieved throughout the doctoral study.

- Giraldo, S.A.C., Supelano, R.C., d'Avila, T.C., Capron, B.D.O., Ribeiro, L.D., Campos, M.M., & Secchi, A.R. (2021). Model Predictive Control with Dead-time Compensation Applied to a Gas Compression System. *Journal of Petroleum Science and Engineering*, 203, 108580. https://doi.org/10.1016/j.petrol.2021.108580

- Giraldo, S.A.C., Supelano, R.C., d'Avila, T.C., Ribeiro, L.D., Campos, M.M., Neto, S.C., & Secchi, A.R. (2020). Controle avançado

de sistemas de compressão de plataformas de produção de óleo e gás natural. In *Rio Oil & Gas 2020: Technical Papers* (No. 181). IBP. https://icongresso.ibp.itarget.com.br/arquivos/trabalhos_-completos/ibp/3/final.IBP0274_20_18112020_133724.pdf

- Giraldo, S.A.C., Melo Junior, P.A., & Secchi, A.R. (2021). Nonlinear Model Predictive Control with Full Dead-Time Compensation. In *Proceedings of the 23rd Brazilian Congress of Chemical Engineering (COBEQ 2021)* (pp. 4). https://proceedings.science/cobeq/cobeq-2021/papers/nonlinear-model-predictive-control-with-full-dead-time-compensation

Chapter 2

Literature Review

This chapter comprehensively reviews the literature on model-based predictive control, focusing on Model Predictive Control (MPC) and Dead Time Compensator (DTC). The chapter is structured into six sections. The first section reviews how the performance of predictive controllers has been assessed in the past, considering approaches based on minimum-variance principles and Linear-Quadratic Gaussian (LQG) benchmarking. The second section introduces the principles of model-based predictive control. The third section examines the challenges and improvements in tuning predictive controllers, including discussions on multi-objective optimization and tuning for non-square systems and robust performance. The fourth section delves into the control of processes with dead time, a key issue in predictive control. The fifth section discusses the detection and diagnosis of predictive controller issues, a crucial aspect of maintaining controller performance, followed by the final remarks.

2.1 Performance Assessment of Predictive Controllers

The control problem in a dynamic system can have diverse goals, such as (i) steady-state regulation, (ii) set-point tracking, and (iii) disturbance rejection. Therefore, tuning a control system must maintain stability and robustness compromises, i.e., sensitivity to changes in the plant parameters, speed of response, and tracking of reference must be considered. The performance of a control system is generally defined by different criteria, which can be divided into the following categories and represented by Figure 2.1 (JELALI, 2013):

- *Deterministic performance criteria.* There are traditional performance measures of the response of a dynamic system, such as the settling time, overshoot, rise time, steady-state error, etc.

- *Stochastic performance criteria.* Typically include the variance, or equivalently, the standard deviation of the controlled variable or control error.

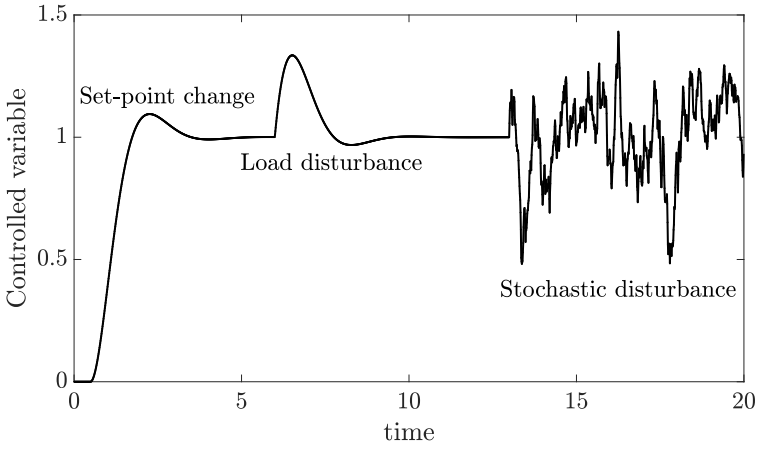


Figure 2.1: Disturbances usually considered for control design.

Such criteria directly relate to process performance, product quality, and energy or material consumption. The steady-state regulation is the essential control problem. Nevertheless, load disturbance rejection response is more critical in closed-loop control, mainly because set-points are fixed in the operating point of the process (SHINSKEY, 1996).

The stochastic criteria for performance assessment in process control usually use the variance expressed in a regulatory control such as:

$$\sigma_y^2 = \frac{1}{N_s - 1} \sum_{k=1}^{N_s} (y(k) - \bar{y})^2, \tag{2.1}$$

where σ_y^2 is the output variance, N_s is the collected output samples, k is the current instant, $y(k)$ is the process output and $\bar{y}$ is given by

$$\bar{y} = \frac{1}{N_s} \sum_{k=1}^{N_s} y(k). \tag{2.2}$$

Combining stochastic and deterministic criteria to guarantee product quality consistency is useful when used for optimal controls such as in MPC. The variance in the output variable of the process is related to product quality, so reducing this variance can improve the final product and make it possible to operate the system near the constraints to increase performance, reduce energy consumption, and save raw materials. This relationship is illustrated in Figure 2.2.

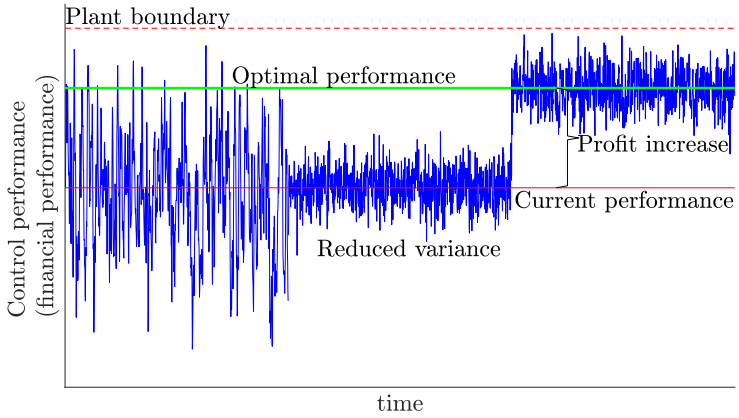


Figure 2.2: Relationship between economic performance and variance reduction.

2.1.1 Assessment Based on Minimum-Variance Principles

Minimum-variance method is concerned with the performance assessment of the control loop with minimum output variance, which does not explicitly consider the control effort. This kind of control has excessive control action and poor robustness, although performance assessment with minimum-variance does provide helpful information as a global lower bound of process variance. In general, tighter quality has, as a result, minor variation in the process output but requires more control effort. So, it is interesting to know how far away the actual control performance is from the best achievable performance with the same control action (HUANG and SHAH, 1999). In mathematical form:

$$\textit{Given that } \mathbb{E}\left\{u^2\right\} \leq \alpha, \textit{ what is the lowest achievable } \mathbb{E}\left\{y^2\right\}? \tag{2.3}$$

Solving the LQG problem with the following objective function,

$$J(\rho_g) = \mathbb{E}\left\{y^2(k)\right\} + \rho_g \mathbb{E}\left\{u^2(k)\right\}, \tag{2.4}$$

where $\mathbb{E}$ is the mathematical expectation, and ρ_g is the move suppression weight. It is possible to obtain the solution of Equation (2.3), i.e., the achievable performance given by the trade-off or Pareto curve, also known as the performance limit curve depicted in Figure 2.3.

Optimal solutions of $\mathbb{E}\left\{y^2(k)\right\}$ and $\mathbb{E}\left\{u^2(k)\right\}$ are found by varying ρ_g. In BOYD and BARRATT (1991), the authors showed that a variety of constraints, such as hard constraints, robustness specification, etc., can be formed as convex optimization problems and are readily solved via convex optimization tools. It is clear that given $\mathbb{E}\left\{u^2(k)\right\} = \alpha$, the minimum value (or the Pareto optimal value) of $\mathbb{E}\left\{y^2(k)\right\}$ can be found from this curve which represents the bound of performance; therefore, it

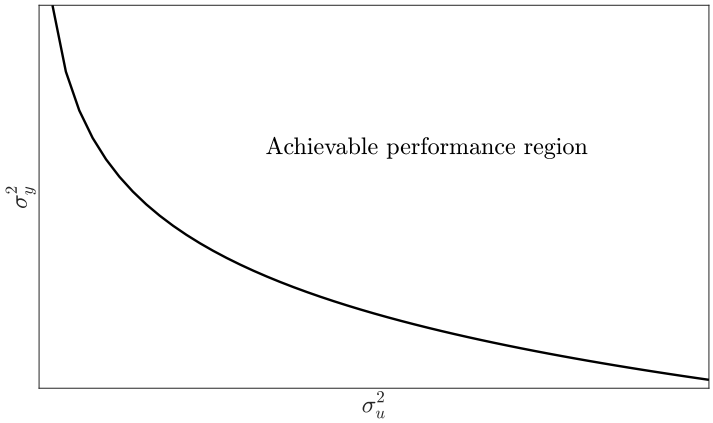


Figure 2.3: An example of the trade-off or Pareto curve that separates achievable and non-achievable performance regions.

can be used for performance assessment purpose.

LQG benchmark can be calculated in different ways as (i) via state space or input-output model, (ii) via generalized predictive control (GPC), and (iii) via the trade-off curve. For a focus on these theories, readers are referred to ÅSTRÖM and WITTENMARK (1990); HUANG and SHAH (1999); KWAKERNAAK and SIVAN (1972). This document instead emphasizes the LQG solution strategy via GPC.

In this section, the focus is predominantly on the utility of minimum-variance methods for control loop performance assessment. While the mathematical rigor presented, specifically through the LQG problem formulation, lends credibility to the arguments, the study reveals limitations in its scope. Primarily, the approach neglects to factor in external noise and real-world constraints, thereby undermining its robustness and applicability. Additionally, although it references alternative LQG benchmark calculations, it falls short of providing a comparative analysis with other methodologies, missing an opportunity to contextualize its findings.

Minimum-Variance Control (MVC)

The minimum variance controller proposed by ÅSTRÖM (1970) aims to reduce the effect of disturbances on the output. It is especially useful when the output of the process to be controlled is contaminated by a stochastic disturbance that cannot be eliminated but can be reduced.

The strategy of this controller is to apply the control signal at the k-time, taking the past values of the input $[u(k-1), u(k-2), ..., u(k-n_b)]$ and of the output $[y(k), y(k-1), y(k-2), ..., y(k-n_a)]$ to minimize a J index.

$$J = \mathbb{E}\left\{y^2(k+d+1|k)\right\}, \tag{2.5}$$

where d is the discrete dead-time and $y^2(k+d+1|k)$ represents the square of the system output at a future time instant $k+d+1$, conditioned on the information available at the current time instant k. This equation indicates that at each sampling period, k, a control signal must be determined to minimize this index J for the subsequent time instant $k+d+1$.

The equation that represents the physical process can be described by an ARMAX model:

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k) + C(z^{-1})v(k), \tag{2.6}$$

where $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ are polynomials in z^{-1} of order n_a, n_b and n_c, respectively:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a}, \\ B(z^{-1}) &= b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + ... + b_{n_b} z^{-n_b}, \\ C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n_c} z^{-n_c}. \end{aligned} \tag{2.7}$$

Polynomial C is filtering the stochastic perturbation $v(k)$ (white noise), and it is added to the system response. Since this polynomial appears in the process model, this term is used as a design parameter of the controller. Polynomial C is one degree smaller than the polynomial A, $n_c = n_a - 1$, to form a causal transfer function. This one is strongly related to an observer polynomial.

To obtain the equation of the model in an instant, $k+d+1$, it suffices to multiply both terms of Equation (2.6) by z^{d+1} and to divide by $A(z^{-1})$:

$$y(k+d+1) = \frac{zB(z^{-1})}{A(z^{-1})}u(k) + \frac{C(z^{-1})}{A(z^{-1})}v(k+d+1). \tag{2.8}$$

Using the division algorithm (i.e., use long division), it can always write the following polynomial identity (WELLSTEAD and ZARROP, 1991):

$$C(z^{-1}) = A(z^{-1})F(z^{-1}) + z^{-(d+1)}G(z^{-1}), \tag{2.9}$$

where:

$$\begin{aligned} F(z^{-1}) &= 1 + f_1 z^{-1} + f_2 z^{-2} + ... + f_d z^{-d}, \\ G(z^{-1}) &= g_0 + g_1 z^{-1} + g_2 z^{-2} + ... + g_{n_a-1} z^{-n_a-1}, \end{aligned} \tag{2.10}$$

it is possible to represent Equation (2.8) in the following way:

$$y(k+d+1) = \frac{zB(z^{-1})}{A(z^{-1})}u(k) + F(z^{-1})v(k+d+1) + \frac{G(z^{-1})}{A(z^{-1})}v(k). \tag{2.11}$$

In the instant k, the last term of Equation (2.11) has the stochastic noise $v(k)$ of the current and the past instants. These terms can be obtained by isolating $v(k)$ from Equation (2.6).

$$v(k) = \frac{A(z^{-1})}{C(z^{-1})}y(k) - z^{-d}\frac{B(z^{-1})}{C(z^{-1})}u(k) \tag{2.12}$$

Substituting Equation (2.12) in Equation (2.11):

$$y(k+d+1) = F(z^{-1})v(k+d+1) + \frac{G(z^{-1})}{C(z^{-1})}y(k) + \frac{zB(z^{-1})F(z^{-1})}{C(z^{-1})}u(k). \tag{2.13}$$

The expression of the output prediction $y(k+d+1)$ is formed by three terms: the first term is formed by future instants not available at the current instant, and the second and third terms have information from the past until the current instant. Therefore, the following expression can be assumed:

$$\hat{y}(k+d+1|k) = \frac{G(z^{-1})}{C(z^{-1})}y(k) + \frac{zB(z^{-1})F(z^{-1})}{C(z^{-1})}u(k), \tag{2.14}$$

then, the best prediction of the output is given by:

$$y(k+d+1) = F(z^{-1})v(k+d+1) + \hat{y}(k+d+1|k). \tag{2.15}$$

This result can be included in the cost function presented in Equation (2.5).

$$J = E\left\{y^2(k+d+1|k)\right\} = E\left\{\left[F(z^{-1})v(k+d+1) + \hat{y}(k+d+1|k)\right]^2\right\} \tag{2.16}$$

From Equation (2.16), it is observed that the cost function is minimized when the output prediction is equal to zero, that is:

$$\begin{aligned}
&\hat{y}(k+d+1|k) = 0 \\
&\frac{G(z^{-1})}{C(z^{-1})}y(k) + \frac{zB(z^{-1})F(z^{-1})}{C(z^{-1})}u(k) = 0 \\
&u(k) = -\frac{G(z^{-1})}{zB(z^{-1})F(z^{-1})}y(k).
\end{aligned} \tag{2.17}$$

This controller cancels the zeros of the system in an open loop due to the presence of polynomial B in the denominator. However, it should be noted that if these zeros are located outside of the unit circle, the system exhibits internal instability in a closed-loop configuration, even in the nominal case. Consequently, any variation in system parameters could exacerbate this unstable behavior.

The Minimum-Variance strategy can be implemented according to Figure 2.4.

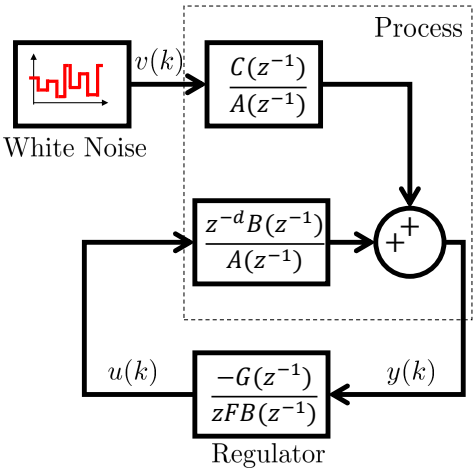


Figure 2.4: MV Regulator.

Example 2.1.1. Consider the following example of a first-order process in difference equation form (WELLSTEAD and ZARROP, 1991)

$$y(k) = 0.8y(k-1) + u(k-1) + v(k) + 0.98v(k-1), \tag{2.18}$$

where $v(k)$ is zero mean white noise of variance $\sigma_e^2 = 1$.

The model in the instant $k + d + 1$ $(d = 0)$ is

$$y(k+1) = 0.8y(k) + u(k) + v(k+1) + 0.98v(k), \tag{2.19}$$

where the prediction model is:

$$\hat{y}(k+1|k) = 0.8y(k) + u(k) + 0.98v(k). \tag{2.20}$$

In this case, the order of the polynomials F and G is zero, therefore the identity from Equation (2.9) takes the form

$$(1 + 0.98z^{-1}) = (1 - 0.8z^{-1})(1) + z^{-1}(g_0), \tag{2.21}$$

leading to $g_0 = 1.78$.

The MV controller is, therefore,

$$u(k) = -1.78y(k), \tag{2.22}$$

note that Equation (2.19) can be written in the form

$$y(k+1) = \hat{y}(k+1|k) + v(k+1), \tag{2.23}$$

so the closed-loop behavior with MV controller of Equation (2.22) is given by

$$y(k+1) = v(k+1). \tag{2.24}$$

Figure 2.5 shows a simulation of the system switching from an open-loop process for $0 \leq k \leq 200$ to MV regulation for $201 \leq k \leq 400$. Note that the output variance as the control changes and under MV regulation, the output $y(k)$ is the same as the corrupting noise $v(k)$.

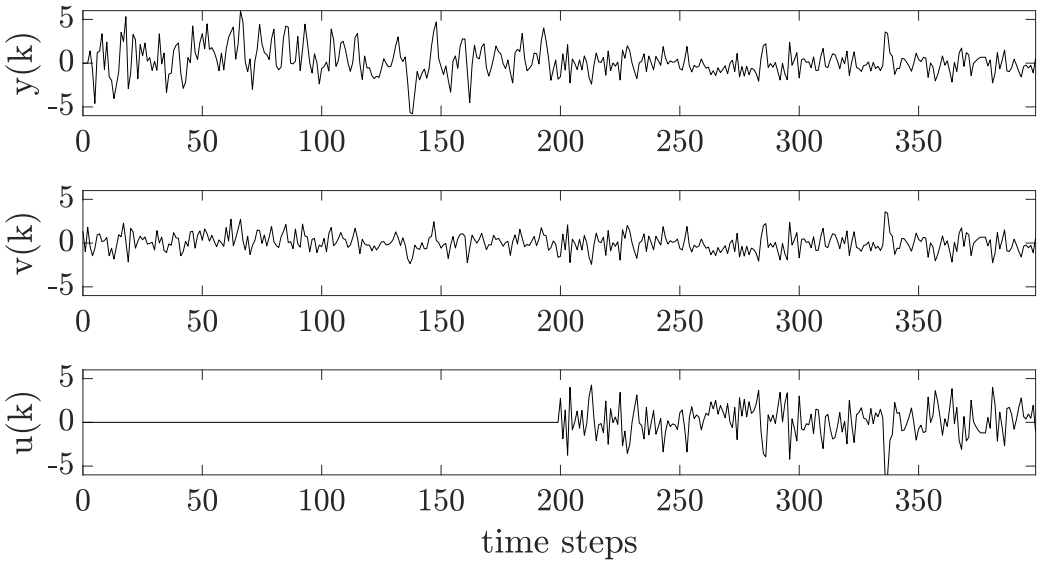


Figure 2.5: Simple MV Regulator.

Minimum-Variance Harris Index

The performance of the MVC-based control loops described by HARRIS (1989) makes a comparison between the output variance of the actual process, σ_y^2, to the MV output variance, σ_{MV}^2, as obtained using the minimum variance controller applied to an estimated time series model from measured output data (JELALI, 2013). Thus, the Harris index is defined as

$$\eta_{MV} = \frac{\sigma_{MV}^2}{\sigma_y^2}, \tag{2.25}$$

with η_{MV} is in the interval $[0, 1]$, and correspond to the ratio of the variance, which could theoretically be achieved under minimum variance control, to the actual variance. Values close to unity indicate good control and close to zero indicate poor performance or instability.

To apply the Harris index on a system, it is necessary to collect the closed-loop data appropriately for the controlled variables and to know or estimate the

dead-time system.

Two important advantages can be highlighted when implementing this index:

1. When taking the ratio between two variances results in a metric that is independent of the underlying disturbances of the system, which is fundamental in industrial processes where there may be variations of disturbances.

2. It is easy to evaluate mainly when it is used in plants with thousands of control loops since it is a non-scale metric in the range of 0 to 1.

While the framework offers two significant advantages—its insensitivity to underlying system disturbances and its scalability for large industrial processes—the study could be critiqued for not addressing potential limitations. For instance, the necessity to know or estimate the dead-time system for applying the Harris Index might introduce an element of approximation error that is not discussed.

Estimation from Time-Series Analysis

Taking the output measurement data of a closed-loop system, it is possible to estimate the time-series using an ARMA model described as (JELALI, 2013):

$$A(z^{-1})y(k) = C(z^{-1})v(k). \tag{2.26}$$

A series expansion, i.e., impulse response (FIR), of this model, gives by:

$$\begin{aligned} y(k) &= \left(\sum_{i=0}^{\infty} e_i z^{-i}\right) v(k) \\ &= \underbrace{\left(e_0 + e_1 z^{-1} + \cdots + e_{d-1} z^{d-1}\right)}_{feedback-invariant} v(k) \\ &\quad + \underbrace{\left(e_\tau z^{-d} + e_{d+1} z^{-(d+1)} + \cdots\right)}_{feedback-varying} v(k). \end{aligned} \tag{2.27}$$

The estimation of the first d coefficients of the impulse response can be calculated through d-term polynomial long division or solve the Diophantine identity given by:

$$\hat{C}(z^{-1}) = \hat{A}(z^{-1})\hat{F}(z^{-1}) + z^{-d}\hat{G}(z^{-1}), \tag{2.28}$$

where $\hat{F}(z^{-1})$ is an estimate of $F(z^{-1})$ in Equation (2.9). The first d feedback-invariant terms only depend on the inherent characteristics of the disturbance that acts on the process and they are not a function of the process model or the controller, Figure 2.6. Therefore, the minimum-variance estimate corresponding to the

feedback-invariant part is given by:

$$\sigma_{MV}^2 = \sum_{i=0}^{d-1} e_i^2 \sigma_v^2. \tag{2.29}$$

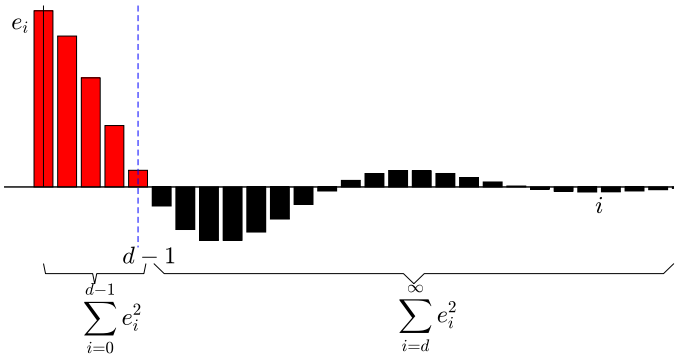


Figure 2.6: An impulse response showing the contributions to the Harris index.

The first coefficient of the impulse response, e_0, is often normalized to be equal to unity.

The output variance can be estimated from the collected output samples using Equation (2.1). Nevertheless, in JELALI (2013), the estimated time series model is suggested to evaluate the current variance. From the series expansion of the time series model from Equation (2.27), the output variance is given by:

$$\sigma_y^2 = \sum_{i=0}^{\infty} e_i^2 \sigma_v^2. \tag{2.30}$$

Note that in Equation (2.25), the noise, $v(k)$, does not affect the performance index since it will be canceled. The index compares the sum of the d first impulse response coefficients squared to the total sum; see Figure 2.6.

In DESBOROUGH and HARRIS (1992), the authors indicated that it is more useful to replace σ_y^2 by the mean-square error of y to account for the offset:

$$\eta_{MV} = \frac{\hat{\sigma}_{MV}^2}{\hat{\sigma}_y^2 + \bar{y}^2}. \tag{2.31}$$

A re-tuning controller is required when η_{MV} is considerably less than 1. If η_{MV} is close to 1, the performance cannot be improved by re-tuning the actual controller, and it is necessary to make process changes to achieve better performance.

While the mathematical rigor and detailed formulations are strengths of this work, there are areas that merit critical evaluation. Firstly, the complexity of the

ARMA models and their associated equations could be computationally intensive, which might be a hindrance in real-time control applications. Secondly, the work suggests replacing σ_y^2 with the mean-square error to account for the offset but does not discuss the implications or the trade-offs involved in this substitution.

2.1.2 Linear-Quadratic Gaussian (LQG) Benchmarking

Considering a cost function of the form:

$$J_{GPC} = \mathbb{E}\left\{\sum_{j=1}^{p}[\hat{y}(k+j|k) - r(k+j)]^2 + \rho_g \sum_{j=1}^{m}[\Delta u(k+j-1)]^2\right\}, \tag{2.32}$$

where $\hat{y}(k+j|k)$ is an optimum j step ahead prediction of the system outputs on data up to time k computed, p is the prediction horizon for process output; m is the control horizon for process input, $r(k+j)$ is a future set-point or reference sequence for the output vector, and $\Delta u(k+j-1)$ is the future incremental control action where $\Delta = 1 - z^{-1}$ is a difference operator (CLARKE *et al.*, 1987).

The control action can be obtained by minimizing the objective function of Equation (2.32), but only the first control action is implemented to obtain a time-invariant control law. For tuning parameters $m = p$, and $p \to \infty$, this objective function converges to the LQG objective function.

KWAKERNAAK and SIVAN (1972) showed that the minimization of this LQG objective function yields a time-invariant optimal control law. Therefore, the LQG problem can be solved via an infinite GPC solution. It is important to note that since p is infinity, GPC computation requires the solution of a large linear problem. Nevertheless, a finite value of the prediction horizon is usually sufficient to achieve the approximate infinite horizon LQG solution via the GPC approach.

2.2 Principles of Model-Based Predictive Control

The process control has been dramatically enhanced by the introduction and use of Model-Based Predictive Control (MPC) systems. These controllers utilize an internal system model to predict future responses and generate appropriate control actions, significantly improving control processes' dynamic performance and stability.

An MPC is a class of advanced control strategies that solves an optimal control problem at every sampling time. The control uses an explicit model to predict the system's outputs at a future time by calculating the future control sequences to minimize a cost function. However, only the first value of the control sequences is applied to the process, and the rest of the predicted future actions are used as

initial guesses for the next optimization cycle. The predictions are adjusted based on new measurements and estimates of the state variables at the next sampling instant, at which point the optimal control problem is solved again (CAMACHO and BORDONS, 2002; CLARKE *et al.*, 1987).

The MPC optimization problem can be described in the classical formulation as:

$$\min_{\mathbf{u},\epsilon_\mathbf{k}} \left\{ J = \sum_{i=1}^{n_y} \sum_{n=1}^{p} \left\{ Q_{ii} \left[\hat{y}_i(k+n|k) - r_i(k+n) \right]^2 \right\} + \sum_{j=1}^{n_u} \sum_{n=1}^{m_j} \left\{ W_{jj} \left[\Delta u_j(k+n-1) \right]^2 \right\} + \rho \epsilon_k^2 \right\}, \tag{2.33a}$$

subject to

$$x(k+n) = f(x(k+n-1),\ u(k+n-1),\ \boldsymbol{\mu}),\ n = 1, \ldots, p \tag{2.33b}$$

$$g(\hat{y}(k+n),\ x(k+n),\ u(k+n),\ \boldsymbol{\mu}) = \mathbf{0},\ n = 1, \ldots, p \tag{2.33c}$$

$$x(k) = \hat{x}(k) \tag{2.33d}$$

$$u_{j_{min}} \leq u_j(k+n-1) \leq u_{j_{max}},\ j = 1, \ldots, n_u \text{ and } n = 1, \ldots, m_j \tag{2.33e}$$

$$\Delta u_{j_{min}} \leq u_j(k+n-1) - u_j(k+n-2) \leq \Delta u_{j_{max}},$$
$$j = 1, \ldots, n_u \text{ and } n = 1, \ldots, m_j \tag{2.33f}$$

$$y_{i_{min}} - \epsilon_k \leq \hat{y}_i(k+n) \leq y_{i_{max}} + \epsilon_k,\ i = 1, \ldots, n_y \text{ and } n = 1, \ldots, p, \tag{2.33g}$$

where J is the cost function; $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{u}$ are vectors of the controlled, state, and manipulated variables, respectively; $\mathbf{r}$ is the reference vector; and $\boldsymbol{\mu}$ is the vector of the model parameters, which can include disturbances. T_s represents the sampling time, which defines the interval at which the control actions are applied. Equations (2.33b) and (2.33c) describe the process model, Equation (2.33d) is the initial condition of the prediction horizon, and Equations (2.33e)–(2.33g) are the lower and upper bounds on the manipulated variables and their rate of variation, and the lower and upper bounds on the controlled variables. n_y and n_u are the numbers of controlled and manipulated variables, respectively, of the MIMO system. p is the prediction horizon value and m_j is the control horizon value of the j-th plant input. All control horizons are condensed in a vector $\mathbf{m}$. $\mathbf{Q}$ and $\mathbf{W}$ are positive semidefinite weight matrices. The future reference trajectory for the i-th plant output at the n-th prediction horizon step is given by $r_i(k+n)$. The predicted value of the i-th plant output at the n-th prediction horizon step is given by $\hat{y}_i(k+n)$, where k is the current control interval. The slack variable ϵ_k is defined for the entire control interval k, a single parameter used throughout the prediction horizon. ρ is

the penalty associated with the violation of soft constraints.

In Equation (2.33), T_s, p, $\mathbf{m}$, $\mathbf{Q}$, and $\mathbf{W}$ are the MPC tuning parameters. Among these, T_s and p are chosen based on the dynamics and performance requirements of the control system, independently of its size. In contrast, $\mathbf{m}$, $\mathbf{Q}$, and $\mathbf{W}$ are parameters that may require more intricate tuning as the size and complexity of the control system increase. According to CAMPOS *et al.* (2013), a poor tuning of the MPC parameters contributes to an ill-posed optimization control problem, which can cause undesired oscillations in the controlled variables. Therefore, each MPC tuning parameter influences the system dynamics (TRIERWEILER and FARINA, 2003). A good selection of the sampling period, control, and prediction horizons allow a good system behavior estimate. It avoids a high computational cost in the calculation of the control action. The weight matrices guarantee the performance and robustness of the system, providing priorities and scaling among the considered process variables.

2.3 Challenges and Improvements in Tuning of Predictive Controllers

The poor performance of the MPC, in addition to the deterioration of the internal model of the controller or the entry of an unmeasured disturbance, can be given to the bad tuning of the controller that eventually does not allow it to reach its control objectives. The tuning of an MPC is not a simple task because of the high number of parameters that increase as the process variables increase.

As part of a lower level in a control function hierarchy, the MPC's role is to minimize the proposed operation objectives, trying to maintain the outputs in the desirable values respecting the constraints, and using the minimal control effort (QIN and BADGWELL, 2003). Different proposals to set the prediction and control horizons and the weighting matrices of the objective function can be found in the literature. Some of these works are presented in a review by RANI and UNBEHAUEN (1997), which addressed tuning methods for Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC) from 1984–1995. Next, in GARRIGA and SOROUSH (2010), a review of the heuristic and theoretical tunings of the MPC method is presented up to the year 2009 and, in ALHAJERI and SOROUSH (2020), a review is presented covering research up to the year 2019.

As indicated in ABRASHOV *et al.* (2017), the classical MPC-tuning procedure did not consider the uncertainties in estimating the MPC parameters. These tunings lack robustness when the system model is not perfectly known. Consequently, model uncertainties provide an error in future output signal estimation. Some authors have

worked in this aspect to improve the behavior of the MPC, within which the works of CHAN and HUA (2015); COETZEE *et al.* (2010); HAN *et al.* (2006) can be cited.

According to GARRIGA and SOROUSH (2010), the MPC tuning strategies, as a general rule, can be divided into two categories: *ad-hoc* and self-tuning. The first one determines the controller parameters by explicit expressions or bounds based on approximation/simulation or parameters of the process dynamic, and the second one sets tuning parameters through optimization algorithms.

There is still no consensus on which methodology provides the best tuning regarding the controller's performance and computational cost. For instance, in SHRIDHAR and COOPER (1997, 1998), *ad-hoc* systematic expressions were employed for tuning the parameters of an MPC based on DMC in both SISO and MIMO systems. Their tuning method was based on a first-order plus dead time (FOPDT) model approximation of the process with zero-order retention. Based on the FOPDT approximation, they obtained an equation to calculate the suppression weights of the controller. The prediction horizon was calculated using an equation based on the condition number of the matrix A of the process, and the control horizon was based on the time constant of the model. TRIERWEILER and FARINA (2003) proposed a robust performance number index to tune the MPC parameters, reflecting the system's directionality and attainable performance to determine the weight matrices.

In TRAN *et al.* (2015), a tuning procedure for GPC was divided into two steps. The first step matched the GPC gain to an arbitrary linear-time-invariant controller (the favorite controller) using the transfer function of the control law. Then, the weight matrices in the cost function were found, resulting in the GPC gain obtained in the first step. A methodology of tuning the control parameters of GPC with long-time-delay plants was proposed in the work of GARCÍA and ALBERTOS (2013). Their methodology used an equivalent representation of the Smith predictor structure. A tuning parameter was provided to reach an intuitive tradeoff between performance and robust stability using sensitivity transfer functions without delay.

Different self-tuning procedures have also been reported in the literature. For instance, in AL-GHAZZAWI *et al.* (2001) expressed a relationship between the MPC controller's parameters and the process outputs through a linear approximation algorithm to analytically obtain the sensitivity functions of $\mathbf{Q}$ and $\mathbf{W}$ in constrained problems. The authors used these functions to guide the MPC feedback response within predefined performance specifications. In VAN DER LEE *et al.* (2008), fuzzy goal programming using the integral square error (ISE) criterion was used to find the MPC tuning parameters through a metaheuristic.

Some contributions were focused on characterizing the set of solutions in the Pareto front, defining different objectives regarding the required specifications of

the process, and solving a multi-objective problem to obtain the tuning parameters of the MPC (REYNOSO-MEZA *et al.*, 2013; VALLERIO *et al.*, 2014). A recursive multi-objective optimization algorithm was presented in WEI LIU and GEORGE WANG (2000), which minimized the sensitivity function between the tuning parameters and the closed-loop performance as the goals of a mixed-integer nonlinear optimization problem. In NERY JÚNIOR *et al.* (2014), an MPC tuning method was proposed using constrained mixed-integer nonlinear programming. In their work, a particle swarm optimization approach was implemented to solve the tuning problem for the worst-case model mismatch scenario, employing a comprehensive combination of the condition number and the Morari resiliency index. In YAMASHITA *et al.* (2016), two optimization algorithms, lexicographic optimization tuning and compromised tuning, were used to obtain the weighting matrices of the MPC control law. In LOZANO SANTAMARÍA and GÓMEZ (2016), a general tuning algorithm for nonlinear model predictive control (NMPC) was presented. The method was based on the utopia tracking concept in a multi-objective optimization problem to adjust the objective function weights. Then, closed-loop performance index optimization was applied to find the horizon lengths. In the work of DE SCHUTTER *et al.* (2020), an open-source software framework was introduced to tune an economic nonlinear model predictive control (ENMPC) process. This tool calculates the optimal stable states or periodic trajectories for constrained nonlinear systems with an economic objective, returning the corresponding positive-definite stage cost matrices for a tracking (N)MPC problem.

Generally, the prediction and control horizons are set following several rules (GARRIGA and SOROUSH, 2010; TRIERWEILER and FARINA, 2003), which have shown satisfactory performance. In the academic literature, selecting single values for control and prediction horizons is common. However, in commercial packages for MPC, the prediction horizon is usually the same for all outputs, although a different control horizon is defined for each input variable. This policy has generally been employed in input blocking design, described in RICKER (1985), and extensively used in industry to reduce computation costs (QIN and BADGWELL, 2003). With this strategy, the control input determined by the optimization stage cannot vary freely at each sampling time of the prediction horizon but only in predefined patterns.

This section has offered an overview of the challenges and methods in the field of MPC tuning, from heuristic approaches to optimization-based strategies. In light of the vast landscape of existing methods, the subsequent chapters will delve into specific tuning strategies highlighted in the literature. Particular attention will be paid to self-tuning methods, which form the core focus of this research. By examining these approaches in detail, this work aims to contribute to the ongoing

discourse of MPC tuning strategies.

2.3.1 Multi-Objective Optimization Technique for Model Predictive Control Tuning: Lexicographic Versus Compromise Approaches

This tuning method was proposed in YAMASHITA *et al.* (2016), where two multi-objective optimizations were implemented to find the weighting matrices of the MPC. The first approach was called as Lexicographic Tuning Technique (LTT) where a lexicographic optimization algorithm is solved to rank the importance of the outputs of the plant. The second one was called as Compromise Tuning Technique (CTT) which uses a compromise optimization approach to solve a multi-objective optimization problem by finding the closest feasible solution to the desired trajectory in terms of the Euclidian distance.

Lexicographic Tuning Technique

This technique arranges the objective function of the MPC in order of importance. This approach is suitable for several formulations of MPC. The authors considered a square system where the number of controlled variables is the same as the number of the manipulated variables to find the MPC parameters in five steps:

1. the user needs to assign the relative importance of the process outputs;
2. an input-output pair is defined for each process output;
3. numerical values of the inputs and outputs, as well as the gains of system model are normalized;
4. the next objective function is minimized:

$$F_i(x) = \sum_{k=1}^{\theta_t} \left|\left| y_i^{ref}(k) - y_i(k) \right|\right|^2, i = 1, 2, ..., n_y, \tag{2.34}$$

where θ_t is the tuning horizon, $y_i^{ref}(k)$ is the discretized reference trajectory of output i, $y_i(k)$ is the closed-loop trajectory of output i, $k = 1, 2, ..., \theta_t$ calculated using the unconstrained version of optimization problem, with objective function given by Equation (2.33); $\mathbf{x_{dv}}$ is the vector of decision variables or tuning parameters, $\mathbf{x_{dv}} = [q_{\delta_1}, ..., q_{\delta_n}, q_{\lambda_1}, ..., q_{\lambda_m}]$, and n is the number of input-output pairs.

5. the Lexicographic optimization is solved:

$$\begin{aligned} \min_{\mathbf{x_{dv}},\delta} V = & \sum_{i=1}^{w'} F_i(x) + \epsilon^T \mathbf{S_t} \epsilon \\ & s.t. \\ & F_i(x) - F_i^* - \epsilon_i \leq 0, i = 1, ..., w' - 1 \\ & \epsilon_i \geq 0, i = 1, ..., w' - 1 \\ & \mathbf{L_B} \leq \mathbf{x} \leq \mathbf{U_B}, \end{aligned} \tag{2.35}$$

where w' is the current tuning step and defines the number of the current output objectives, ϵ is a vector of slack variables, $\mathbf{S_t}$ is a diagonal weighting matrix, $\mathbf{L_B}$ and $\mathbf{U_B}$ are the lower and upper bounds of the decision variables and F_i^* is the optimum value of the objective goal (utopia solution).

Compromise Tuning Technique

In this technique, the tuning goals are defined in the same way as LTT, where the utopia solution is obtained by the following optimization problem:

$$\begin{aligned} F_i^*(x) = & \min_x F_i(x), \ i = 1, ..., w \\ & s.t. \\ & \mathbf{L_B} \leq \mathbf{x} \leq \mathbf{U_B}. \end{aligned} \tag{2.36}$$

With the utopia solution, the CTT try to find the closest feasible solution to it, in terms of the Euclidian distance.

$$\begin{aligned} \min_x \ & ||\mathbf{F}^*(x) - \mathbf{F}(x)||^2 \\ & s.t. \\ & \mathbf{L_B} \leq \mathbf{x} \leq \mathbf{U_B}. \end{aligned} \tag{2.37}$$

These two tuning techniques are defined in terms of output reference trajectories, describing the desired time-domain characteristics. As can be observed in the LTT method, the degrees of freedom depend on the number of system inputs to define the goals of the objective function and is more suitable for square processes. The CTT is independent of the size of the system because it can take into account as many objectives as necessary. In these methods, the selection of the Pareto optimum solutions is done automatically once the tuning goals are specified.

2.3.2 Tuning Strategy for Non-Square Systems and Range Controlled Variables Based on Multi-Scenarios Approach for Model Predictive Control

In different engineering branches, such as chemical engineering, it is common to find non-square systems with more outputs than inputs. Therefore, the system must operate in zones using the soft-constraint operation to attend to the control specifications (LIMA and GEORGAKIS, 2006).

SANTOS *et al.* (2017) proposed a tuning strategy based on decomposing the non-square system in square sub-scenarios with two or more channels using all permutations of possible combinations of inputs and outputs.

Usually, the original system tends to be reduced to the square concerning available manipulated and active controlled variables, which violates the soft-constraints. Nevertheless, a smaller system can bring performance limitations not seen in the full problem. Therefore, it is necessary to calculate the best attainable performance function for all scenarios to determine the system's adequate scaling and the tuning matrices' weights.

Since a non-square system is considered, the control of the variables is done by zones, and every submodel may have attainable or unattainable constraints that need consideration for the closed-loop performance. Only the important scenarios are chosen using segmentation techniques, such as operating frequency. This method determines the attainable closed-loop transfer function matrix for the whole process model, fulfilling the attainable constraints present in each submodel originating from the complete model. A desirable deterministic performance is defined to determine the smallest achievable value for the rise time, respecting the control action limitation and considering maximal sensitivity robustness criterion using the following objective function:

$$\begin{aligned} \min_{\mathbf{\Omega}} \phi(\Omega) = & \min_{\mathbf{\Omega}} \sum_{i=1}^{n_u} (rt_i)^2 \\ & s.t \\ & rRPN(\kappa) \leq 1 \\ & Ms(\kappa) \leq 2.2 \\ & Ms(\kappa) \geq 1.2 \\ & Mk(\kappa) \leq 10 \end{aligned} \tag{2.38}$$

where rt is the desired rise time, $\mathbf{\Omega} = [rt_1, rt_2, ..., rt_i]$ with $i = 1...n_u$ being the number of outputs of the complete transfer matrix, $\kappa = 1...nc$, the number of possible square subsystems originated from the full model, $rRPN(\kappa)$ the relatively robust

performance number of the scenario κ, $Ms(\kappa)$ corresponds to the maximal sensitivity value and $Mk(\kappa)$ to the maximal value of the function $K(s)$ that relates to $\Delta u = K(s)\Delta y_{set}$ for each scenario, where Δy_{set} is the set-point variation.

The $rRPN(\kappa)$ is obtained as a deviation between the area under the curve of the function Γ (A_T) and the area under the curve of the function Γ_{REF} (A_{REF}).

$$rRPN = \frac{A_T - A_{REF}}{A_{REF}} \tag{2.39}$$

Γ is the Robust performance function given by:

$$\Gamma(\mathbf{G}, \mathbf{T}, \omega) = \sqrt{\bar{\sigma}([\mathbf{I} - \mathbf{T}(j\omega)]\mathbf{T}(j\omega)) \left[\beta^*(\mathbf{G}(j\omega)) + \frac{1}{\beta^*(\mathbf{G}(j\omega))}\right]}, \tag{2.40}$$

$$RPN = \Gamma_{max}(\mathbf{G}, \mathbf{T}, \omega) = \max\left[\Gamma(\mathbf{G}, \mathbf{T}, \omega)\right], \tag{2.41}$$

where $\beta^*(\mathbf{G}(j\omega))$ is the minimum condition number of $\mathbf{G}(j\omega)$, $\bar{\sigma}([\mathbf{I} - \mathbf{T}(j\omega)]\mathbf{T}(j\omega))$ is the maximal singular value of the transfer function matrix $[\mathbf{I} - \mathbf{T}(j\omega)]\mathbf{T}(j\omega)$, and $\mathbf{T}(s)$ is the attainable closed-loop transfer function for the model $\mathbf{G}(s)$ and the controller $\mathbf{C}(s)$.

The function Γ_{REF} represents the minimal possible value of the function Γ for a determined desirable performance and the corresponding minimum condition number of the transfer function matrix. The functions Γ and Γ_{REF} are illustrated in Figure 2.7.

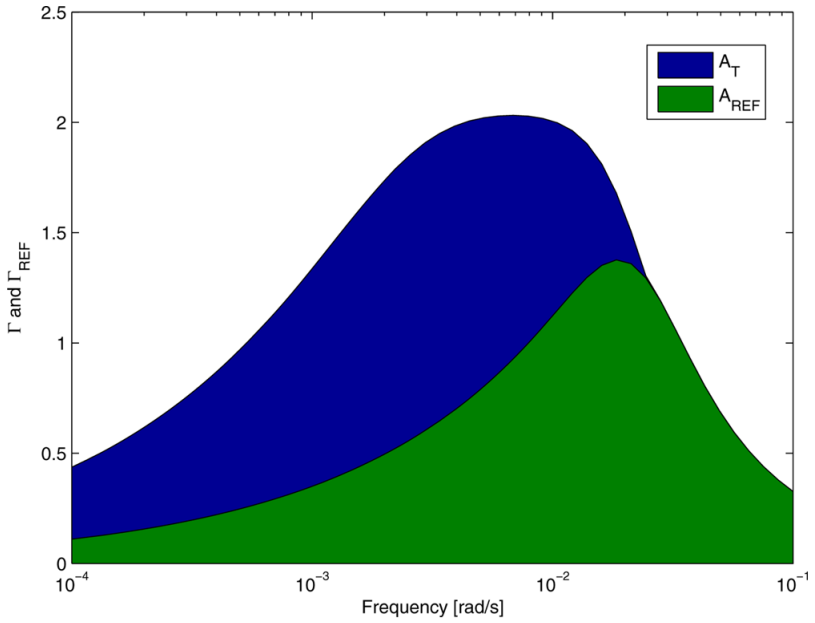


Figure 2.7: Γ and Γ_{REF} functions (SANTOS *et al.*, 2017).

This optimization aims to obtain the faster desired closed-loop performance function, respecting the attainable constraints existing in each scenario. The system is scaled using as a reference the frequency was found through the following optimization function

$$\beta^* = \min_{\mathbf{L},\mathbf{R}} ||\beta_\kappa(\mathbf{LG}(i\omega_{Max}, \kappa)\mathbf{R})||_\infty, \tag{2.42}$$

where $\mathbf{R}$ and $\mathbf{L}$ are diagonal scaling matrices for inputs and outputs, respectively, $\mathbf{G}(i\omega_{Max}, \kappa)$ is the original model of the process evaluated in the frequency ω_{Max}, κ where the RPN of the attainable trajectory for each scenario κ occurs, $\beta(\mathbf{G})$ is the condition number of the matrix $\mathbf{G}$, and $||\mathbf{x}||_\infty$ is the infinity norm of $\mathbf{x}$ (LIMA and GEORGAKIS, 2006).

The tuning controller parameters are found to minimize the following optimization problem, where the decision variables, $\mathbf{x} = [\epsilon, \mathbf{Q}, \mathbf{W}]$, correspond to the weighting matrices $\mathbf{Q}, \mathbf{W}$ and ϵ, which will penalize the move suppression, the error between the outputs and the set-point signal, and the soft-constraint violation, respectively.

$$\min_{\mathbf{x}} \psi(\mathbf{x}) = \min_{\mathbf{x}} \left[\sum_{\kappa=1}^{nc} \left\| T_\kappa(s)\Delta y_{set} - \underbrace{(\hat{y} - y_{bias})}_{\Delta\hat{y}} \right\|^2 + \sum_{\kappa=1}^{nc} \hat{\epsilon}_\kappa^2 \right], \tag{2.43}$$

where $T_\kappa(s)$ is the attainable closed-loop transfer function for the system, $\hat{y}$ is the MPC simulated outputs, and $\hat{\epsilon}_\kappa$ is the maximal violation of the soft-constraint given by the controlled variables using the scaled system and controller based on the model $\mathbf{G}$.

2.3.3 Tuning of MPC for Robust Performance

The modeling of industrial processes is strongly linked to different types of uncertainties, making the closed-loop robustness of an MPC control system indispensable when using a tuning technique to find its parameters. In the literature, it is possible to find different MPC tuning strategies. However, few of them present a robustness guarantee that assures the algorithm's practical implementation.

Some authors have presented robust MPC schemes to consider model uncertainty explicitly, such as in SCOKAERT and MAYNE (1998), which used the min-max optimization problem, presented in Equation (2.44), to find the proper trajectory of the input control when the system is operating in the worst condition, where θ is the magnitude of model uncertainty. This approach improved the performance compared to the standard MPC scheme and avoided the feasibility problems that result from using the min-max formulation because it includes the notion that feed-

back is present. Nevertheless, the computational demands of the feedback min-max algorithms can be very high.

$$\min_{\Delta u} \max_{\theta} \sum_{j=1}^{p} ||\hat{\mathbf{y}}(k+j|k) - \mathbf{r}(k+j)||^2_{\mathbf{Q}} + \sum_{j=1}^{m} ||\mathbf{\Delta u}(k+j-1)||^2_{\mathbf{W}} \tag{2.44}$$

Tuning Strategy based on Min-Max optimization

In the work of HAN *et al.* (2006) is presented a robust tuning strategy based on Min-Max optimization using Particle Swarm Optimization (PSO) to deal with the complex optimization problem. This strategy tries to solve the following objective function:

$$\begin{aligned} &\min_{\mathbf{P_C}} \max_{\mathbf{P_G}} EISTWE(\mathbf{P_C}, \mathbf{P_G}) \\ &s.t. \\ &\mathbf{P_C} \in S \\ &\mathbf{P_G} \in U, \end{aligned} \tag{2.45}$$

where $\mathbf{P_C} = [p, m, \lambda_1, ..., \lambda_{n_u}]$, $\mathbf{P_G}$ is a vector that contains the model parameters such as the gain, the time constant, the delay and the sampling period; U is a close set which represents the variation range of model parameters relating to the model uncertainty θ; S is a predefined problem space; the $EISTWE$ is a performance index giving by:

$$EISTWE = \left(1 + \frac{\lambda\ (a+b)}{Setpoint}\right) \int_0^{\infty} \left(t^n e(t)^2 + \delta \Delta u(t)^2\right) dt + \rho_r\ m\ p, \tag{2.46}$$

where $Setpoint$ is the final value of $r(t)$, $e(t)$ is the difference between reference trajectory and process output, $e(t) = r(t) - y(t)$, $\Delta u(t)$ denotes the move of manipulated variables; λ, δ, and ρ_r are weights, a is the maximum desirable overshoot and b is the first decay peak of a typical overdamped second-order response.

Faster time response is forced by t^n multiplier, where n is fixed at 1, 2, or 3. The term $\delta \Delta u(t)^2$ avoids sharp and large moves in the manipulated variables.

In the $EISTWE$ index, a combination of a proper p and a proper smaller m is made, so the last term of Equation (2.46) is to avoid large p and m being selected.

While the methodology employs the EISTWE performance index to favor the selection of parameters that enhance robustness and reduce real-time computational burden, a more extensive validation across different systems and operational conditions is warranted. Although the acquired parameters suggest reduced computational overhead, a rigorous assessment of computational efficiency in broader sce-

narios would further solidify its contributions. The strategy offers valuable insights into the design of robust MPC controllers; however, additional research is needed to confirm its applicability and efficiency across various systems.

Tuning MPC with Anticipation

ABRASHOV *et al.* (2017) presented a robust MPC tuning method fitting the controller directly in the anticipative mode where the future reference is known.

Initially, the authors established a desired closed-loop transfer function to achieve a tuning that approximates this dynamic, so the closed-loop transfer function is given by:

$$\mathbf{G}_{CL}^{ref}(z^{-1}) = \frac{\mathbf{R}_B(z^{-1})}{\mathbf{R}_A(z^{-1})} z^{-1}, \tag{2.47}$$

where $\mathbf{R}_B(z^{-1})$ and $\mathbf{R}_A(z^{-1})$ are the polynomials of the numerator and denominator of degree n_{rb} and n_{ra} respectively with corresponding coefficients $\mathbf{r}_b, \mathbf{r}_a$.

In scenarios where future reference signals are not known, an alternative approach is used. The MPC's actual closed-loop transfer function, $\mathbf{G}_{CL}(z^{-1})$, is derived based on the following expression:

$$\mathbf{G}_{CL}(z^{-1}) = \frac{\mathbf{B}_{CL}(z^{-1})}{\mathbf{A}_{CL}(z^{-1})} z^{-1}, \tag{2.48}$$

In this equation, $\mathbf{B}_{CL}(z^{-1})$ and $\mathbf{A}_{CL}(z^{-1})$ are the numerator and denominator polynomials of the actual MPC closed-loop transfer function. The coefficient vectors for these polynomials are represented by $\mathbf{b}$ and $\mathbf{a}$, respectively, where to ensure the same performance of the reference transfer function, it must be ensured that the following function is fulfilled:

$$f_i(\mathbf{x}) = 0, \;\; \mathbf{f}(\mathbf{x}) = [\mathbf{b} - \mathbf{r}_b, \mathbf{a} - \mathbf{r}_a], i = 1, ..., L, \tag{2.49}$$

where L is the length of vector $\mathbf{f}$. If coefficient vectors are of different sizes, the smaller one is filled with zeros for missing polynomial degrees. To ensure the exact solution of Equation (2.49), it is necessary to formulate the following optimization problem:

$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}} \;\; \epsilon^T \epsilon \\ s.t. \;\; &-\epsilon \leq f_i(\mathbf{x}) \leq \epsilon, \;\; i = 1, ...L \end{aligned} \tag{2.50}$$

Solving the optimization problem given by Equation (2.50) permits the matching of the closed-loop transfer functions in Equations (2.47) and (2.48). Nevertheless,

to avoid numerical issues, the problem can be solved in the frequency domain by replacing $\mathbf{f}(\mathbf{x})$ in Equation (2.49) by $\mathbf{w}(\omega) = \mathbf{G}_{CL}^{ref}(\omega) - \mathbf{G}_{CL}(\omega)$ with $\mathbf{G}_{CL}^{ref}(\omega)$, $\mathbf{G}_{CL}(\omega)$ which are the frequency domain representations of $\mathbf{G}_{CL}^{ref}(z^{-1})$, $\mathbf{G}_{CL}(z^{-1})$ respectively.

When the future reference is considered, the MPC closed-loop transfer function is given by:

$$\mathbf{y}(z) = \frac{\mathbf{G}_{CL}(z)}{\kappa_s}\left(k_1 r_1(z) + k_2 r_2(z) + \ldots + k_p r_p(z)\right) \tag{2.51}$$

where $\kappa_s = \sum_{i=1}^{p} k_i$, k_i are the elements of the vector $\mathbf{K}_1$, and $r_i(z)$ are the corresponding reference signal elements. The vector $\mathbf{K}_1$ represents the first row of the optimal gain matrix $\mathbf{K}$, which is analytically obtained by solving the objective function of the MPC. Specifically, the optimal gain matrix $\mathbf{K}$ is computed as:

$$\mathbf{K} = (\mathbf{G}^T\mathbf{Q}\mathbf{G} + \mathbf{W})^{-1}\mathbf{G}^T\mathbf{Q} \tag{2.52}$$

where $\mathbf{G}$ is the coefficient matrix, $\mathbf{Q}$ is the reference tracking weighting matrix, and $\mathbf{W}$ is the control increment weighting matrix.

The following optimization problem is solved to minimize the variance of gain vector $\mathbf{K}_1$ and obtain a prescribed controller behavior while the reference varies and consider that the initial system has M parametric states:

$$\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}} \left[\alpha\epsilon^T\epsilon + (1-\alpha)\left(\mathbf{K}_1 - \frac{\kappa_s}{p}\right)^T\left(\mathbf{K}_1 - \frac{\kappa_s}{p}\right)\right] \\ s.t. \quad & -\epsilon \leq f_{i,j}(\mathbf{x}) \leq \epsilon,\ i = 1,...L,\ j = 1,...M \\ & \mathbf{f}_j(\mathbf{x}) = [\mathbf{b}_j - \mathbf{r}_{b_j}, \mathbf{a}_j - \mathbf{r}_{a_j}] \end{aligned} \tag{2.53}$$

where α is a weighting coefficient.

For the robustness problem, this approach considers that the closed-loop performance could be achieved for each M parametric state of the system. A parametric state refers to a specific set of operating conditions or configurations characterized by a particular set of parameters. Therefore, any robust controller structure can be implemented to achieve closed-loop performance for each parametric state. The corresponding transfer functions for each parametric state are $\mathbf{G}_{CL_1}^{ref}(z), \ldots, \mathbf{G}_{CL_M}^{ref}(z)$. MPC closed-loop transfer functions are also calculated for each parametric state and denoted as $\mathbf{G}_{CL_1}(z), \ldots, \mathbf{G}_{CL_M}(z)$ using the optimization problem presented in Equation (2.53).

Although the mathematical formulation of this method enhances the MPC robustness, it is imperative to consider certain limitations. The computational com-

plexity associated with the optimization problems remains undetermined, which could impede real-time applications, especially for systems with multiple parametric states. Furthermore, the authors themselves highlight the restrictive nature of their parametrization via the weighting matrices $\mathbf{Q}$ and $\mathbf{W}$, posing potential challenges for controller adaptability. The method's performance under stochastic or uncertain conditions is not exhaustively discussed. Thus, while contributing significantly to robust MPC design, the method raises questions that warrant further investigation.

2.3.4 Analysis of MPC Controller Tuning Methods

The scientific literature proposes various methodologies for optimizing the tuning parameters of MPC. Notable among these are LTT and CTT, which rely on predefined reference trajectories and employ optimization algorithms to adjust system parameters (as mentioned in previous sections). While these methods have been successful in achieving the desired process behavior, it is important to scrutinize their limitations. Such trajectory-based methods are inherently susceptible to model inaccuracies and disturbances, thereby raising questions about their robustness. Additionally, these algorithms often result in locally optimal solutions, posing the risk of suboptimal system-level performance.

The advent of tuning methods for non-square systems has piqued interest, especially due to its applicability to complex chemical engineering processes. Despite the constraints in degrees of freedom that these systems present, working in zones, as discussed in section 2.3.2, appears to be a viable approach. Yet, this approach does not come without its challenges. The robustness during transitions between different zones and the implications for system stability remain critical and unresolved issues.

It is paramount to note the significance of robust tuning methodologies, such as those presented in section 2.3.3. Traditional tuning often relies heavily on aggressive adjustments based on the internal model of the controller. Such aggressive tuning, although effective under certain conditions, exposes the system to instability risks in the presence of model-plant mismatches.

In conclusion, while the existing tuning methods offer an array of benefits, none appear to be universally applicable to all types of systems. Each approach comes with its own set of limitations and challenges that must be carefully weighed. The observations indicate a manifest need for future research focused on more versatile and robust tuning strategies.

2.4 Control of Processes with Dead Time

Most chemical industrial processes have dead times (or time delays) in their dynamics. Typical examples of industrial processes are distillation columns and heat exchangers. Dead time can be caused, for example, by the time required to transport mass, energy, or information, by the accumulation of time lag in a great number of low-order systems connected in series, and even by the necessary time to perform the algorithm control law (NORMEY-RICO and CAMACHO, 2007).

Two control structures are widely used in industry as ways to address the problems caused by dead-time in the closed-loop system: dead-time compensator (DTC) and model-based predictive control (MPC). DTCs use a prediction of the process output that allows, under some circumstances, a controller to be designed to the process as if it were delay-free (NORMEY-RICO and CAMACHO, 2008). MPC also uses predictions, but in this case, the control signal is obtained as the solution of an optimization problem at each iteration (CAMACHO and BORDONS, 2002).

The forthcoming sections discuss the fundamental principles and mechanisms of the DTC and MPC control structures for systems with dead time are explored. The aim is to provide a comprehensive understanding of how these strategies can effectively tackle the challenges of dead times in controlling chemical industrial processes.

2.4.1 Smith Predictor

In 1957, the North-American engineer Otto J. M. Smith proposed the most popular dead-time compensation algorithm, known as Smith predictor (SP) (SMITH, 1957). This algorithm uses a linear model identified from the real process to predict the future behavior of the process output.

The control structure is shown in Figure 2.8. In the figure, $P_n(s) = G_n(s)e^{-L_n s}$ is the nominal model of the process, $G_n(s)$ is the model without transport delay, known in the literature as a fast model, L_n is the transport delay of the nominal model, $C(s)$ is the primary controller, $P(s)$ is the real plant, $u(t)$ is the control action, $y(t)$ is the process variable, $q(t)$ is the load disturbance, $n_1(t)$ is the output disturbance, $n_2(t)$ is the measured noise, $e_p(t)$ is the predicted error, $r(t)$ is the reference trajectory, $e(t)$ is the control error and $f(t)$ is the predicted output.

This structure predicts the output of the real process, $y(t)$, based on the fast model, $G_n(s)$, predicting the behavior of the system as if it had no transport delay, $e^{-L_n s}$. Thus, the control can anticipate a probable behavior of the system, $P(s)$, in a time equal to the transport delay (ZHONG, 2006). Based on this structure, it is possible to design the primary controller, $C(s)$, considering the process without delay.

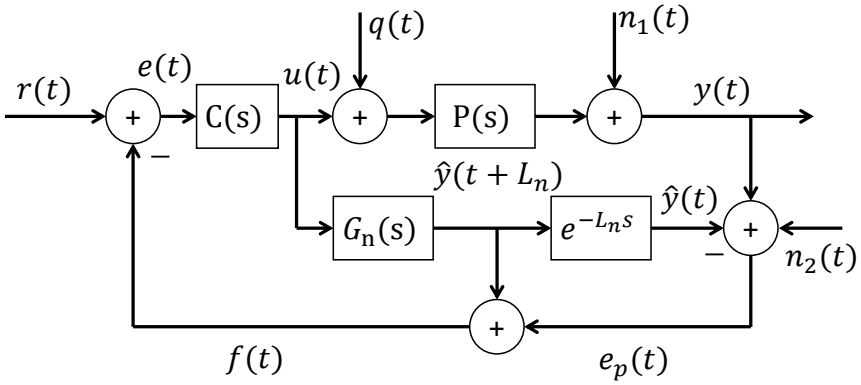


Figure 2.8: Smith predictor.

Several SP-based control strategies have emerged in recent years to mitigate the key limitations associated with SP control (PALMOR, 1996). These limitations include its inapplicability to unstable systems, slow disturbance dynamics, and challenges in dealing with dead times in the fast model of MIMO systems. While it has been traditionally thought that dead times cannot be entirely eliminated in MIMO systems, this notion is not universally valid. As outlined in Chapter 11 of NORMEY-RICO and CAMACHO (2007), the feasibility of dead time removal is dependent on specific design steps and the characteristics of the system in question. Therefore, the difficulty in primary controller design varies according to these factors.

2.4.2 Filtered Smith Predictor

As a result of the inherent limitations of the original Smith Predictor, numerous modifications and tuning procedures have been proposed over the past four decades. These improvements aim to enhance performance under measurable and non-measurable disturbances, control integrating or unstable plants, and increase robustness or simplify tuning. Though generally more complex than the original Smith Predictor and particularly designed for specific types of processes, these control techniques are extensively reviewed in relevant literature (NORMEY-RICO and CAMACHO, 2008; PALMOR, 1996).

A unified SISO Dead-Time Compensator design approach has been suggested in NORMEY-RICO and CAMACHO (2009), known as the Filtered Smith Predictor (FSP). Capable of controlling stable, integrating, and unstable dead-time processes, this novel controller allows for unified design and tuning.

The FSP structure to control a process with dominant time delay is depicted in Fig 2.9. $P(s)$ is the plant in continuous time, ZOH is a zero-order hold, T_s is the sampling period, $F_r(z)$ is the predictor filter, $C(z)$ is the primary controller, $G_n(z)$ is the fast model, $P_n(z) = G_n(z)z^{-d_n}$ is the nominal model (discretized with a zero-

order hold), d_n is the nominal time delay. Moreover, $r(k)$ is the reference trajectory, $e(k)$ is the control input, $u(k)$ is the control action, $y(k)$ is the output, $\hat{y}(k)$ is the model output, $y_p(k)$ is the output prediction, $q(k)$ is the load disturbance, and $n(k)$ is an output disturbance.

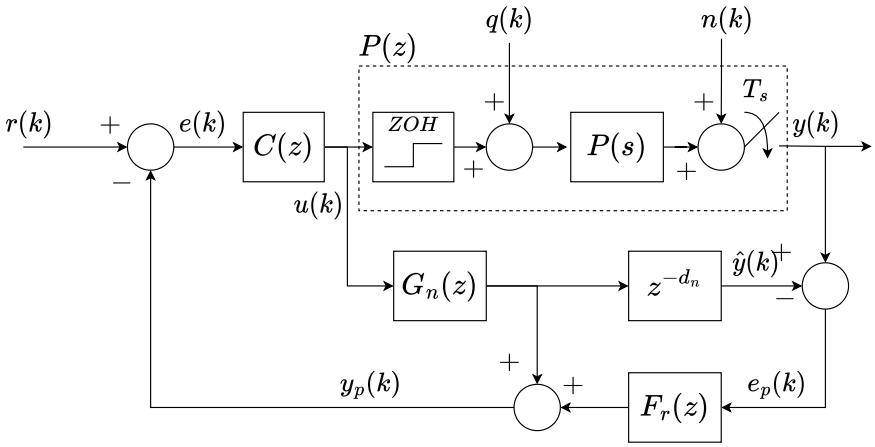


Figure 2.9: The Filtered Smith Predictor (FSP).

The predictor filter, $F_r(z)$, allows adjusting the disturbance rejection response and noise attenuation, which increases the robustness and guarantees internal stability of the control structure (FLESCH *et al.*, 2011). The prediction error, $e_p(k)$, is zero when the model is perfect, and the structure works appropriately for the stable process. However, when there is a modeling error in the time delay, the signal $e_p(k)$ starts to oscillate because of the lag of the signals $y(k)$ and $\hat{y}(k)$, which may lead the closed-loop system to instability. Therefore, the control structure has improved its response by designing the filter to attenuate these oscillations, making it more robust.

In the case that there is no modeling error, i.e., $P(z) = P_n(z)$, the closed-loop transfer function is represented by Equation (2.54), by Equation (2.55) for load disturbance, and by Equation (2.56) for output disturbance:

$$H_{yr}(z) = \frac{y(k)}{r(k)} = \frac{C(z)P_n(z)}{1 + C(z)G_n(z)}, \tag{2.54}$$

$$H_{yq}(z) = \frac{y(k)}{q(k)} = P_n(z)\left[1 - \frac{C(z)P_n(z)F_r(z)}{1 + C(z)G_n(z)}\right]. \tag{2.55}$$

$$H_{yn}(z) = \frac{y(k)}{n(k)} = \left[1 - \frac{C(z)P_n(z)F_r(z)}{1 + C(z)G_n(z)}\right]. \tag{2.56}$$

From Equations (2.54) to (2.56), it is possible to see that the filter $F_r(z)$ has no effect on the reference track but can be used to change the disturbance rejection

responses in case slow, integrating, or unstable poles appear. The structure of Figure 2.9 is only used for analysis and cannot be used for implementation in the unstable case since internal instability occurs when the load disturbance appears. For this reason, it has to be undertaken to implement the unified structure presented in Figure 2.10.

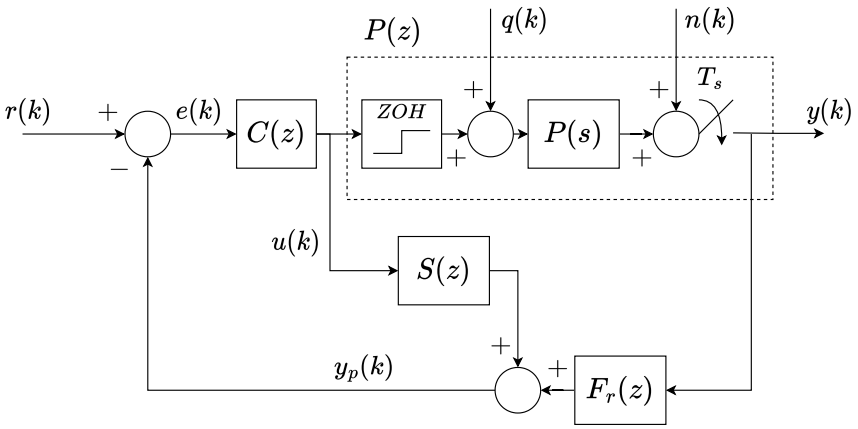


Figure 2.10: Unified Filtered Smith Predictor.

The $S(z)$ block guarantees the internal stability of the structure and is defined by:

$$S(z) = G_n(z)\left[1 - z^{-d_n}F_r(z)\right]. \tag{2.57}$$

For the new structure to be internally stable, it must be guaranteed that the $S(z)$ filter and the $F_r(z)$ filter are stable. In the case of the $F_r(z)$ filter, it is easy to guarantee its stability since its poles are chosen freely by the designer. On the other hand, in the $S(z)$ filter, given by Equation (2.57), it can be noted that if $G_n(z)$ has unstable poles, these poles only can be canceled by the $F_r(z)$ filter avoiding the internal instability. For this reason, the term $\left[1 - z^{-d_n}F_r(z)\right]$ must always be calculated in a way that cancels the unstable dynamics of the fast plant model, $G_n(z)$. In this way, the predicted output $y_p(k)$ will be a stable prediction of $y(k)$.

2.4.3 Robustness of Filtered Smith Predictor

Small modeling errors in the control structure can lead the system to instability. This problem can be solved with a proper choice of the $F_r(z)$ filter. The robustness analysis of the system in the presence of modeling errors is typically done by representing the system through a family of models, $P_i(z)$, rather than representing the plant with a linear model. Among the models belonging to the family, there is a model called the nominal model, $P_n(z)$, which is usually employed to represent the standard or average behavior of the system. This type of representation is

an unstructured uncertainty description, presented in Equation (2.58), which represents the unmodeled dynamics by the transfer functions, nonlinearities, and effects derived from linearization processes. A significant advantage of this procedure is that it allows a simple and unified representation of uncertainties regardless of their origin (NORMEY-RICO and CAMACHO, 2007).

$$P_i(z) = P_n(z)\left[1 + \delta P_i(z)\right] = P_n(z) + \Delta P_i(z), \tag{2.58}$$

with:

$$|\delta P_i(z)| \leq \overline{\delta P_i}(z), \tag{2.59}$$

where $\delta P_i(z)$ defines the frequency domain error shape for each model i, while $\overline{\delta P_i}(z)$ is its limiting modulus. The frequency domain representation is obtained by replacing z with $e^{j\omega T_s}$, where T_s is the sampling period and the term $e^{j\omega T_s}$ is periodic in frequency with a period of $\frac{2\pi}{T_s}$. From Equation (2.58), it can also be observed that:

$$\delta P_i(z) = \frac{P_i(z) - P_n(z)}{P_n(z)}. \tag{2.60}$$

The characteristic equation of the closed-loop system for FSP, defined for $P_i(z) \neq P_n(z)$, is described by:

$$1 + C(z)G_n(z) + C(z)F_r(z)\left[P_i(z) - P_n(z)\right] = 0. \tag{2.61}$$

Thus, to ensure that all the roots of Equation (2.61) are inside the unit circle, the robust stability condition is given by Equation (2.62) (NORMEY-RICO and CAMACHO, 2007).

$$\overline{\delta P_i}(e^{j\omega T_s}) < dP(e^{j\omega T_s}) = \left|\frac{1 + C(e^{j\omega T_s})G_n(e^{j\omega T_s})}{C(e^{j\omega T_s})G_n(e^{j\omega T_s})F_r(e^{j\omega T_s})}\right|, \quad 0 < \omega < \pi/T_s, \tag{2.62}$$

where $dP(e^{j\omega T_s})$ is the robustness index considering the maximum permissible modeling error magnitude for each frequency to maintain closed-loop stability. From Equation (2.62), it is possible to see that $F_r(z)$ aims to reduce the gain of $dP(e^{j\omega T_s})$ in frequency regions where multiplicative uncertainties threaten the robust stability condition. Qualitatively, it means attenuating modeling error expression in the feedback signal.

In addition to the above discussion, a MATLAB function is also available in the following repository for calculating and graphing the robustness index of the Filtered Smith Predictor structure. This function takes the process transfer function, the model transfer function, the controller, and the robustness filter as inputs:

https://github.com/sergioacg/MATLAB-Functions/blob/main/robutness.m.

This tool can be a great aid in analyzing the robustness of a control system, especially when dealing with Filtered Smith Predictors. By utilizing this function, it is possible to visualize the impact of modeling errors on system stability, aiding in selecting an appropriate robustness filter. The graphed output can give insights into the frequency regions where multiplicative uncertainties pose the most significant threat to the robust stability condition.

2.4.4 Model Predictive Control based in a Dead-Time Compensator

Traditional predictive controllers often have low robustness when the plant to be controlled has a significant dead time compared to other dynamics involved. As a way around this problem, CAMACHO and BORDONS (2002) and NORMEY-RICO and CAMACHO (2007) showed that a prediction structure based on the FSP could be used in unison with the predictive controller, in this case with the generalized predictive control (GPC), forming a dead-time compensator generalized predictive controller (DTC-GPC) structure. This uses the robustification of control systems for dead-time processes from DTC structure (FSP) into the MPC framework to facilitate the robust design of the GPC control.

Consider that the process can be represented by a MIMO discrete $n_y \times n_u$ transfer function $\mathbf{P}(z^{-1})$ with sampling period T_s.

$$\mathbf{y}(k) = \mathbf{P}(z^{-1})\mathbf{u}(k), \tag{2.63}$$

where the variables $\mathbf{y}(k)$ and $\mathbf{u}(k)$ are the $n_y \times 1$ output and the $n_u \times 1$ input vectors and each element $p_{ij}(z^{-1})$ of $\mathbf{P}(z^{-1})$ is a SISO transfer function:

$$p_{ij}(z^{-1}) = \frac{z^{-1}B^*_{ij}(z^{-1})}{A_{ij}(z^{-1})} z^{-d_{ij}}, \tag{2.64}$$

with d_{ij} as a dead time of the transfer function between the j-th input and i-th output expressed as a number of sampling times where $B^*_{ij}(z^{-1})$ and $A_{ij}(z^{-1})$ are polynomials in the backshift operator.

The effective dead time of the i-th output is $d_i = \min_j[d_{ij}]$, which allows the MIMO model to be described as:

$$\hat{\mathbf{P}}(z^{-1}) = \mathbf{D}(z^{-1})\mathbf{A}^{-1}(z^{-1})\mathbf{B}(z^{-1})z^{-1}, \tag{2.65}$$

where $\mathbf{D}(z^{-1})$ is a polynomial diagonal matrix with elements z^{-d_i}; $\mathbf{A}(z^{-1})$ is a diagonal polynomial matrix with elements $A_i(z^{-1})$ equal to the least common multiple

of the denominators $A_{ij}(z^{-1})$ of the corresponding row of matrix $\hat{\mathbf{P}}(z^{-1})$; $\mathbf{B}(z^{-1})$ is a polynomial matrix with elements $B_{ij}(z^{-1})$ such that

$$g_{ij}(z^{-1}) = \frac{z^{-1}B_{ij}(z^{-1})}{A_i(z^{-1})}, \tag{2.66}$$

that is, $g_{ij}(z^{-1})$ are the SISO transfer functions without the common delay d_i of the corresponding output. The polynomials $A_i(z^{-1})$ and $B_{ij}(z^{-1})$ are given by

$$\begin{aligned} A_i(z^{-1}) &= 1 + a_1^i z^{-1} + a_2^i z^{-2} + ... + a_{n_{a_i}}^i z^{-n_{a_i}}, \\ B_{ij}(z^{-1}) &= b_1^{ij} + b_2^{ij} z^{-1} + b_3^{ij} z^{-2} + ... + b_{n_{b_{ij}}}^{ij} z^{-n_{b_{ij}}}. \end{aligned} \tag{2.67}$$

The DTC-GPC algorithm applies a control sequence that minimizes the multistage cost function presented in Equation (2.33). The prediction of the system output is computed using the following controlled auto-regressive integrated moving average (CARIMA) model of the plant:

$$\mathbf{A}(z^{-1})\mathbf{y}(k) = \mathbf{D}(z^{-1})\mathbf{B}(z^{-1})\mathbf{u}(k) + \frac{1}{\Delta}\mathbf{T}(z^{-1})\mathbf{e}(k), \tag{2.68}$$

where $\mathbf{T}(z^{-1})$ is an $n_y \times n_y$ monic polynomial matrix representing the colouring polynomials of the noise, but in this work, it is considered as identity matrix, $\Delta = 1 - z^{-1}$. The variable $\mathbf{e}(k)$ is the $n_y \times 1$ noise vector.

Since $\mathbf{A}(z^{-1})$ is diagonal, it is possible to use a Diophantine equation in the following ARIMAX model to obtain the optimal predictions of each n_y output of the system as shown below:

$$A_i(z^{-1})y_i(k) = z^{-d_i}\mathbf{B}_i(z^{-1})\mathbf{u}(k-1) + \frac{1}{\Delta}e(k), \tag{2.69}$$

where $\mathbf{B}_i(z^{-1}) = [B_{i1}, B_{i2}, ..., B_{in_u}]$. The polynomial identity (CAMACHO and BORDONS, 2002)

$$1 = E_{ij}(z^{-1})\Delta A_i(z^{-1}) + z^{-j}F_{ij}(z^{-1}), \tag{2.70}$$

allows obtaining the following optimal prediction of the system:

$$\hat{y}_i(k+j|k) = F_{ij}(z^{-1})\hat{y}_i(k) + z^{-d_i}E_{ij}(z^{-1})\Delta\mathbf{B}_i(z^{-1})\mathbf{u}(k-1), \tag{2.71}$$

where $E_{ij}(z^{-1})$ is of order $j-1$ and $F_{ij}(z^{-1})$ of order n_{ai}.

The DTC-GPC splits the prediction procedure of the output of the plant into two parts. The first one, the prediction of the output of the plant up to dead time, is computed using an FSP structure. The second one uses an MPC structure based

on the CARIMA model for forecasting the behavior of the real process, from dead time plus one period of time to the defined horizon.

The MIMO Filtered Smith predictor is an extension that uses the same structure of the SISO version of FSP, and its block diagram is shown in Figure 2.11. In the figure, $\mathbf{P}(s)$ is the plant with multiple time delays, $\mathbf{G_n}(z)$ is its fast model, $\mathbf{L}(z)$ is the nominal MIMO delay model, $\mathbf{P_n}(z) = \mathbf{L}(z)\mathbf{G_n}(z)$ is the nominal process model (discretized with a zero-order hold), $\mathbf{F_r}(z)$ is the MIMO predictor filter, and $\mathbf{C}(z)$ is a MIMO primary controller. All discrete-time signals are represented as functions of the variable k, meaning a multiple of the sampling period T_s. Moreover, $\mathbf{r}(k)$ is the reference trajectory, $\mathbf{u}(k)$ is the control action, $\mathbf{y}(k)$ is the output, $\widehat{\mathbf{y}}(k)$ is the model output, $\mathbf{e_p}(k)$ is the prediction error, and $\mathbf{n}(t)$ is an output disturbance.

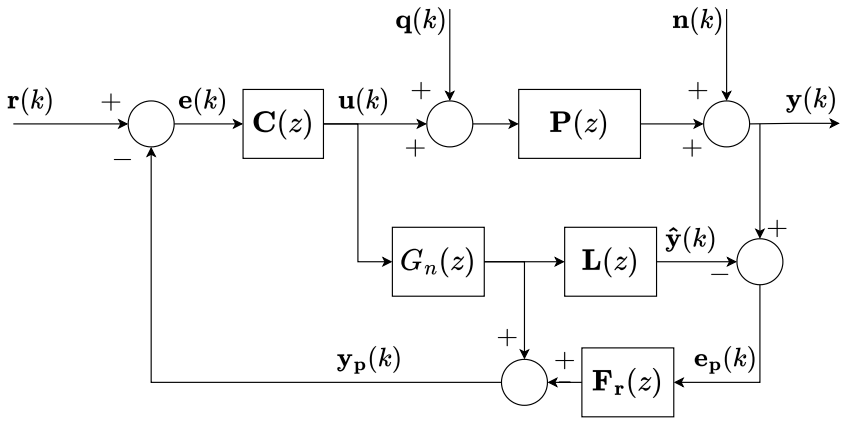


Figure 2.11: MIMO-FSP scheme structure.

From Figure 2.11 of the MIMO FSP, it is possible to obtain the output prediction equation of the system until the dead time, which is given by:

$$\hat{\mathbf{y}}(k + d_i|k) = \mathbf{G_n}(z^{-1})\mathbf{u}(k) + \mathbf{F}_r(z^{-1})\left[\mathbf{y}(k) - \mathbf{P_n}(z^{-1})\right], \tag{2.72}$$

where $\mathbf{F_r}(z^{-1})$ is a matrix of diagonal transfer functions and $\hat{\mathbf{y}}(k + d_i|k)$ is the prediction vector.

It is possible to calculate the predictions, from the dead time plus one period of time to the prediction horizon, that are used in the cost function J of Equation (2.33) as a function of a free response $\mathbf{f_r}$ (which depends on past actions) and the forced response of the system (which depends on future actions). For this, if Equation (2.71) is applied recursively, the output prediction of DTC-GPC control can be reduced to the following matrix expression:

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{u_f} + \mathbf{H_p}\mathbf{u_p} + \mathbf{S}\hat{\mathbf{y}}_\mathbf{p}, \tag{2.73}$$

where $\mathbf{H}\mathbf{u_f}$ is known as forced response and $\mathbf{f_r} = \mathbf{H_p}\mathbf{u_p} + \mathbf{S}\hat{\mathbf{y}}_\mathbf{p}$ is a free response of

the system. And:

$$\begin{aligned}
\hat{\mathbf{y}} &= [\hat{\mathbf{y}}_\mathbf{1}, \hat{\mathbf{y}}_\mathbf{2}, ..., \hat{\mathbf{y}}_\mathbf{ny}] \\
\mathbf{u_f} &= [\mathbf{u_{f_1}}, \mathbf{u_{f_2}}, ..., \mathbf{u_{f_{n_u}}}] \\
\mathbf{u_p} &= [\mathbf{u_{p_1}}, \mathbf{u_{p_2}}, ..., \mathbf{u_{p_{n_u}}}] \\
\hat{\mathbf{y}}_\mathbf{p} &= [\hat{\mathbf{y}}_{\mathbf{p_1}}, \hat{\mathbf{y}}_{\mathbf{p_2}}, ..., \hat{\mathbf{y}}_{\mathbf{p_{n_u}}}] \\
\mathbf{H} &= \begin{bmatrix} \mathbf{H_{11}} & \mathbf{H_{12}} & \cdots & \mathbf{H_{1n_u}} \\ \mathbf{H_{21}} & \mathbf{H_{22}} & \cdots & \mathbf{H_{2n_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H_{n_y1}} & \mathbf{H_{n_y2}} & \cdots & \mathbf{H_{n_yn_u}} \end{bmatrix},
\end{aligned} \tag{2.74}$$

where the future control actions are given by

$$\mathbf{u_{f_j}} = \begin{bmatrix} \Delta u_j(k) & \Delta u_j(k+1) & \cdots & \Delta u_j(k+m_j) \end{bmatrix}^T, \; m_j = \min_i(p_i - d_{ij}), \tag{2.75}$$

and the past control actions are

$$\mathbf{u_{p_j}} = \begin{bmatrix} \Delta u_j(k-1) & \Delta u_j(k-2) & \cdots & \Delta u_j(k-n_{b_{ij}}) \end{bmatrix}^T. \tag{2.76}$$

Each block $\mathbf{H_{ij}}$ of the matrix $\mathbf{H}$ is calculated using the step response of the subsystem i, j, that is, considering the polynomials $A_i(z^{-1})$ and $B_{ij}(z^{-1})$.

The matrix $\mathbf{H_p}(z^{-1})$ has $\sum_i^{n_y} p_i$ rows and n_u columns and each element $H^i_{p_{kj}}(z^{-1})$ is a polynomial with the same order as $B_{ij}(z^{-1})$. Here, i corresponds to the output, j to the input and k to the instant considered on the horizon, thus, $H^i_{p_{kj}}(z^{-1})$ gives the relation between $\Delta u_j(k)$ and $\hat{y}_i(k + d_i + 1|k)$.

Because of the independence between the predictions, $\mathbf{S}$ is a block-diagonal matrix, each block being of dimension $p_i \times n_{a_i}$.

The optimum of Equation (2.33) can be expressed as

$$\mathbf{u} = \mathbf{K}(\mathbf{r} - \mathbf{f_r}), \tag{2.77}$$

where $\mathbf{r}$ represents the future reference and $\mathbf{K}$ is given by

$$\mathbf{K} = (\mathbf{H}^T\mathbf{Q}\mathbf{H} + \mathbf{Q})^{-1}\mathbf{H}^T\mathbf{W}. \tag{2.78}$$

Only the first n_u rows of $\mathbf{K}$ (defined as $\mathbf{K}_{n_u}$) have to be computed. $\mathbf{K}_{n_u}$ has p_w columns, where $p_w = p_1 + p_2 + ... + p_{n_y}$, which corresponds to the number of variables in the horizon.

Now, using the expression of $\mathbf{f_r}$

$$\Delta\mathbf{u}(k) = \mathbf{K}_{n_u}\mathbf{r} - \mathbf{K}_{n_u}\mathbf{H_p}\mathbf{u_p} - \mathbf{K}_{n_u}\mathbf{S}\hat{\mathbf{y}}. \tag{2.79}$$

For a focus on this MPC algorithm, readers are referred to NORMEY-RICO and CAMACHO (2007), and an example is presented in Appendix B.

As shown in this section, the dead time in chemical processes represents a difficulty in the control systems design. This difficulty increases in the MIMO systems due to different delays in the internal interactions of the system variables.

2.5 Detection and Diagnosis of Predictive Controller Issues

Predictive controllers, although robust and effective in handling multivariate control problems, are not immune to performance degradation due to issues such as model-plant mismatches, unmeasured disturbances, and inaccuracies in the internal model. These issues can significantly affect the predictive capability of the controller, leading to suboptimal performance or even instability. This section reviews the existing literature on the detection and diagnosis of such issues within predictive controllers. Emphasis is placed on the strategies and techniques developed to identify and address the source of performance degradation to maintain the optimal performance of the control system.

2.5.1 Model Predictive Control Assessment

The essence of a predictive control is based on the dynamic model of the process, because through this, the controller manages to predict the future outputs considering a control action which is optimized throughout the control horizon. This optimization is calculated at each sampling time according to the receding horizon concept of the MPC. This optimization problem considers the slack on the process constraints, the weight matrices, the control and prediction horizons as tuning parameters. Therefore, it is possible to observe that in the literature there is no consensus that specifies which is the best solution for performance assessment of the MPC. This difficulty of monitoring and diagnosing in the MPC structure is a direct consequence of the algorithm complexity. Without a doubt, the deterioration of the dynamic model of the process is one of the causes of poor performance of the MPC controller, nevertheless, finding this type of degradation is not a simple task. In addition, the existence of unmeasured disturbances increases this difficulty (BOTELHO *et al.*, 2016b).

Several techniques have emerged in industrial and academic processes about the MPC assessment. Some of these techniques use the minimum variance controller (MVC) and its extension, the linear quadratic Gaussian (LQG). For example, in LEE *et al.* (2008) the authors evaluated the sensitivity of the process variables using the MVC method to determine which one has a greater economic impact. The effect of constraints on the quality of the controller operation was proposed in HARRISON and QIN (2009) using a minimum variance map based on LQG. Some methods are briefly discussed in the following.

2.5.2 Constrained Minimum-Variance Control

Minimum-variance control is a frequent benchmark for feedback control performance assessment, especially due to its non-intrusiveness and ability to provide the absolute lower bound of process variance. Supposing that the process model has a stable inverse, the constrained minimum variance control approach, proposed in KO and EDGAR (2001), consists of designing a constrained MVC using the receding horizon concept of the MPC. Thereupon, it is necessary to estimate the achievable MV performance bounds in constrained MPC, via disturbance model identification and closed-loop simulation using the constrained MVC (JELALI, 2013).

The drawback of the minimum variance benchmark, however, is an aggressive control and the requirement of the interactor matrix that is conceptually difficult and computationally challenging (HUANG and KADALI, 2008).

2.5.3 Infinite-Horizon Model Predictive Control

Controlled auto-regressive integrated moving average (CARIMA) model is typically used in the MPC structure to obtain the control law presuming that unmeasured disturbance is a random signal, i.e.

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k) + \frac{C(z^{-1})}{\Delta}v(k). \tag{2.80}$$

The objective of the MPC is to minimize the cost function of Equation (2.32) using the tuning presented in section 2.1.2 to converge to the LQG objective function. This solution is known as the infinite-horizon MPC controller.

As the system model is known, because it is explicitly integrated into the design of a model predictive controller, the MPC performance limit curve can be constructed as LQG curve by plotting σ_y^2 vs $\sigma_{\Delta u}^2$ with $\rho_g \in [0, \infty)$ and $\sigma_{\Delta u}^2$ as the process input variance. Moreover, the infinite-horizon MPC controller has to be expressed in the Reference Signal Tracking (RST) form to compute the curve. For more details, readers are referred to JELALI (2013).

It has been shown in JULIEN *et al.* (2004) that under certain circumstances it is possible to identify a system model from normal operating data to update the internal model of the MPC. A new performance curve can be obtained using the old controller and the new process model (OCNP). With this curve, it is possible to re-tune the existing controller to achieve desirable performance region despite the presence of MPM eliminating the need for a new identification experiment.

Also, it is possible to design a new MPC controller based on the new model (NCNP) to generate a performance curve. The distance between the NCNP curve and the OCNP curve indicates the performance deficiency due to the MPM, see Figure 2.12.

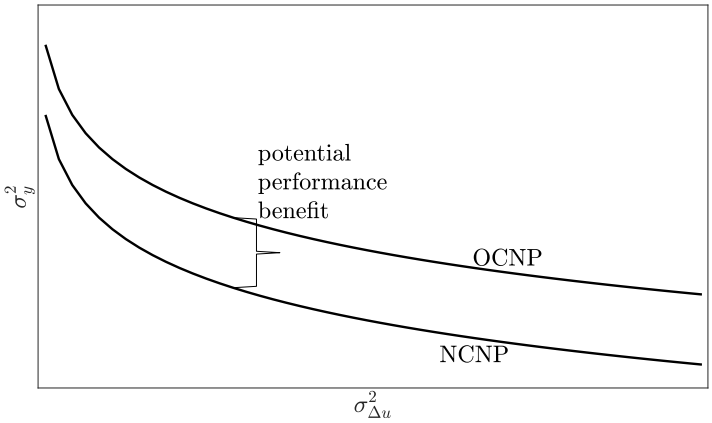


Figure 2.12: MPC performance curves.

When the controller is re-tuned varying the move suppression coefficient, ρ_g, the performance shows the OCNP curve. Hence, the OCNP and NCNP curves enable to differentiate between variance inflation caused by tuning vs MPM. The potential performance benefit of re-identifying the plant can be justified against the time and expense associated with a new response test.

While the study is rigorous in theoretical formulation, certain caveats must be considered. First, although the text alludes to re-tuning of the controller to improve performance, it does not delve into the computational complexity or practical feasibility of this re-tuning process. Second, it remains unclear how the study accommodates non-linearities or uncertainties intrinsic to real-world systems.

2.5.4 Performance Monitoring of Model-Predictive Controllers via Model Residual Assessment

Proposed in SUN *et al.* (2013), this technique focuses on evaluating MPC model quality. It is based on residual assessment and feedback invariant principles, whereby

disturbance innovations are not affected by the feedback controller. Using the set-points, denoted as $\mathbf{r}(k)$, and the measured outputs $\mathbf{y}(k)$, the estimation of stochastic disturbance error, $\mathbf{e^d}(k)$, is conducted according to:

$$\begin{aligned}\mathbf{y}(k) &= \sum_{i=1}^{\infty} \mathbf{A}(z^{-1})\mathbf{y}(k-i) + \sum_{i=1}^{\infty} \mathbf{B}(z^{-1})\mathbf{r}(k-i) + \mathbf{e^d}(k) \\ &\approx \sum_{i=1}^{M_1} \mathbf{A}(z^{-1})\mathbf{y}(k-i) + \sum_{i=1}^{M_2} \mathbf{B}(z^{-1})\mathbf{r}(k-i) + \mathbf{e^d}(k),\end{aligned} \tag{2.81}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the parameters of the ARX model, and M_1 and M_2 are the model orders.

Assume that the plant and disturbance model used in MPC can be represented as:

$$\mathbf{y}(k) = \mathbf{P_n}(z^{-1})\mathbf{u}(k) + \mathbf{Q_0}(z^{-1})\mathbf{v}(k), \tag{2.82}$$

where $\mathbf{P_n}(z^{-1})$ is the process model and $\mathbf{Q_0}(z^{-1})$ is the disturbance model. The prediction error, $\mathbf{e^p}(\mathbf{k})$, is determined based on one-step-ahead prediction (LJUNG, 1999), being the optimal prediction of the output using past measurements. The prediction error is represented by:

$$\hat{\mathbf{y}}(k|k-1) = \mathbf{Q_0}(z^{-1})^{-1}\mathbf{P_n}(z^{-1})\mathbf{u}(k-1) + (\mathbf{I} - \mathbf{Q_0}(z^{-1})^{-1})\mathbf{y}(k-1), \tag{2.83}$$

$$\mathbf{e^p}(k) = \hat{\mathbf{y}}(k|k-1) - \mathbf{y}(k), \tag{2.84}$$

In scenarios where the disturbance model is not present, LJUNG (1999) suggests employing a filter (predictor) to encapsulate the disturbance impact, given by $\mathbf{Q_0}(z^{-1})\mathbf{e^d}(k)$. The author recommends a performance indicator, MQI, expressed as:

$$MQI = \frac{\sum_{i=1}^{n_s} \mathbf{e^d}(k)^T \mathbf{Q} \mathbf{e^d}(k)}{\sum_{i=1}^{n_s} \mathbf{e^p}(k)^T \mathbf{Q} \mathbf{e^p}(k)}, \tag{2.85}$$

where $\mathbf{Q}$ are controlled variables weights in MPC controller and n_s is the number of sampled data. The index MQI varies between 0 and 1. When this index is close to 1, it means that the error is due to stochastic disturbance and the model is perfect (BOTELHO *et al.*, 2016b).

While the study presents a methodologically sound approach, it assumes that the disturbance model is perfectly known or encapsulated by a filter, which might not hold true in practice. Furthermore, the utilization of the MQI index, although promising, lacks empirical validation in real-world scenarios. This raises questions regarding the scalability and robustness of the method when subject to model-plant mismatches or unforeseen disturbances.

2.5.5 Detection of Model-Plant Mismatch in MPC applications

This methodology, proposed in BADWE *et al.* (2009), is based on the analysis of partial correlations between the model residuals and the manipulated variables. The use of partial correlation is necessary to avoid false detection of model mismatches given by the causal relation between the variables.

First, it is necessary to choose a period of data of an output error model, $\mathbf{e^p}(k)$, (the difference between output process and output model) and manipulated variables with sufficient set-point excitation in the process.

Then, the effect of each manipulated variable is isolated according to

$$\hat{u}_i^r(k) = \mathbf{G_{u_i}}\tilde{\mathbf{u}}^\mathbf{r}(k) + \epsilon_{u_i}, \tag{2.86}$$

where $\hat{u}_i^r(k)$ is the evaluated manipulated variable, $\mathbf{G_{u_i}}$ is the evaluated model, $\tilde{\mathbf{u}}^\mathbf{r}$ contains all manipulated variables except $\hat{u}_i^r(k)$, and ϵ_{u_i} is that component of $\hat{u}_i^r(k)$ that is uncorrelated with the manipulated variables in $\tilde{\mathbf{u}}^\mathbf{r}$. Component ϵ_{u_i} can be estimated as

$$\hat{\epsilon}_{u_i}(k) = \hat{u}_i^r(k) - \mathbf{G_{u_i}}\tilde{\mathbf{u}}^\mathbf{r}(k). \tag{2.87}$$

The term $\hat{\epsilon}_{u_i}$ is that component of the manipulated variables that is free of effect from disturbances and the other manipulated variables. Similarly, it is necessary to isolate the effect of each controlled variables from all manipulated variables, except the one being evaluated. For this, it is calculated the output error model between the prediction error evaluated from controlled variables and the remaining manipulated variables.

$$e_j^p(k) = \mathbf{G_{e_j}}\tilde{\mathbf{u}}^\mathbf{r}(k) + \epsilon_{e_j}(k), \tag{2.88}$$

and obtain an estimate of $\epsilon_{e_j}(k)$,

$$\hat{\epsilon}_{e_j}(k) = e_j^p(k) - \mathbf{G_{e_j}}\tilde{\mathbf{u}}^\mathbf{r}(k). \tag{2.89}$$

Model mismatch is detected through the regular correlation of $\hat{\epsilon}_{u_i}$ and $\hat{\epsilon}_{e_j}$. A non-zero correlation between these terms indicate the presence of model-plant mismatch in the $u_i - y_j$ channels. The more significant this correlation is, the more significant model-plant mismatch is (BADWE *et al.*, 2009).

A unique aspect of this study is the quantifiable detection of MPM via the correlation of isolated variables, which offers a numerical metric for the severity of the mismatch. However, the methodology presupposes the availability of data with sufficient set-point excitation and makes assumptions about the separability

and independence of manipulated variables. Moreover, the study does not address potential challenges linked to computational complexity or real-time application constraints.

2.5.6 Diagnosis of Unmeasured Disturbance versus Model–Plant Mismatch in MPC

BADWE *et al.* (2010) consider an internal model control (IMC) structure presented in Figure 2.13 to analyze the MPC performance, where $P(z)$ is the plant, $C(z)$ is the IMC controller, $P_n(z)$ is the nominal model. Moreover, $r(k)$ is the reference trajectory, $e(k)$ is the error, $u(k)$ is the control action, $u_d(k)$ is the designed control action, $y(k)$ is the output, $y_d(k)$ is the designed output, $\hat{y}(k)$ is the model output, $y_p(k)$ is the output prediction, $n(k)$ is an output disturbance.

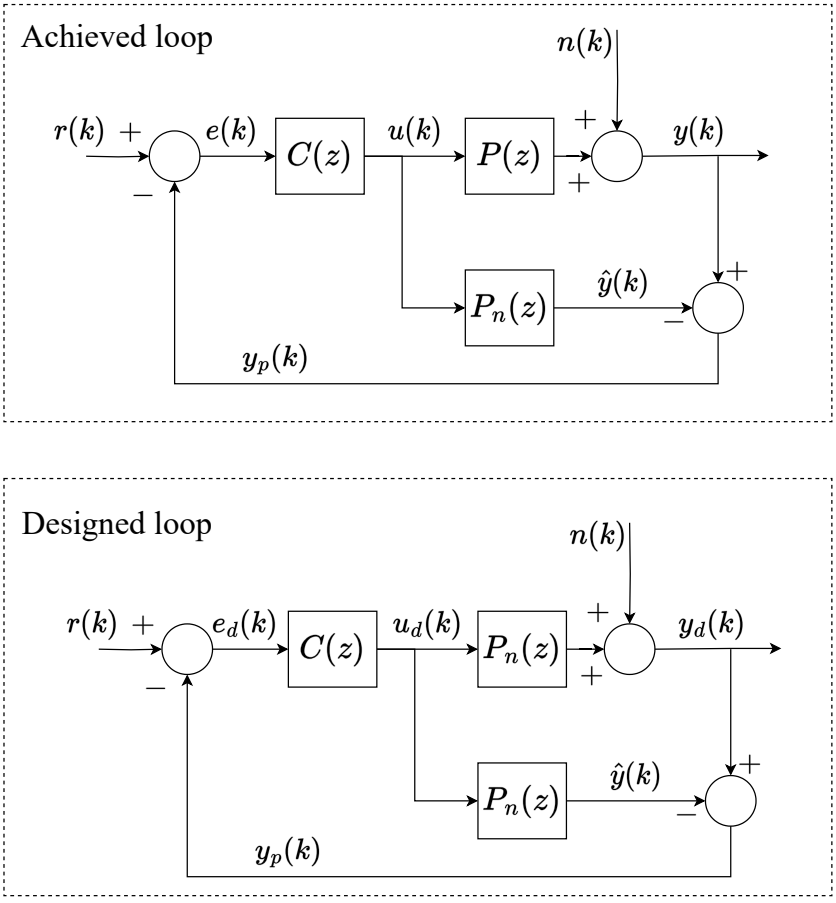


Figure 2.13: IMC structure for the achieved and designed control loops.

In the designed loop, the assumption is made that the model is a precise representation of the real plant, i.e., $P(z) = P_n(z)$. Under this assumption, the output prediction $y_p(k)$ equals the output disturbance $n(k)$, and the desired output $y_d(k)$ can be calculated as $y_d(k) = C(z)P_n(z)e_d(k) + n(k)$.

Given that the error design signal $e_d(k)$ can be written as the difference between the reference signal and the output disturbance, $e_d(k) = r(k) - n(k)$, because $P(z) = P_n(z)$, so it is possible to substitute this into the equation for $y_d(k)$. This is given by:

$$y_d(k) = C(z)P_n(z)r(k) + \Big(\underbrace{1 - C(z)P_n(z)}_{S_d(z)}\Big)n(k), \tag{2.90}$$

where $S_d(z)$ is the closed-loop relationship between the controlled variable and the disturbances known as designed sensitivity.

The function $S_d(z)$ can be determined in several ways. Depending on the intricacy of the controller in use, $S_d(z)$ could be derived analytically or alternatively identified through an experimental procedure.

In the identification process, the variables $y(k)$, $r(k)$, and $y_p(k)$ are considered. The control error $e(k)$ is given by $e(k) = r(k) - y(k) + \hat{y}(k)$, where $\hat{y}(k) = C(z)P_n(z)e(k)$. Inserting this into the original equation yields $e(k) = r(k) - y(k) + C(z)P_n(z)e(k)$. Rearranging terms, $(1 - C(z)P_n(z))\, e(k) = r(k) - y(k)$. Lastly, it is possible to express $r(k) - y(k)$ in terms of $S_d(z)$ and $e(k)$:

$$r(k) - y(k) = S_d(z)\Big(\underbrace{r(k) - y_p(k)}_{e(k)}\Big). \tag{2.91}$$

The variability gives a controller performance degradation factor on the controlled variables (MPM effect). Therefore, the authors proposed to compare the variability observed in the control loop with the variability that would have been obtained in the case of the perfect model under the same conditions. To obtain the nominal or design output, $y_d(k)$, it is necessary to initially estimate the loop gain, $C(z)\Delta P(z)$, where $\Delta P(z)$ is the MPM given by $P(z) - P_n(z)$. So, the loop gain can be identified by an output-error model using process data as $y_p(k) = C(z)P(z)e(k) - C(z)P_n(z)e(k)$, so:

$$y(k) - \hat{y}(k) = C(z)\Delta P(z)e(k), \tag{2.92}$$

where $e(k)$ can also be written as $e(k) = r(k) + C(z)P_n(z)e(k) - C(z)P(z)e(k) - n(k)$, then:

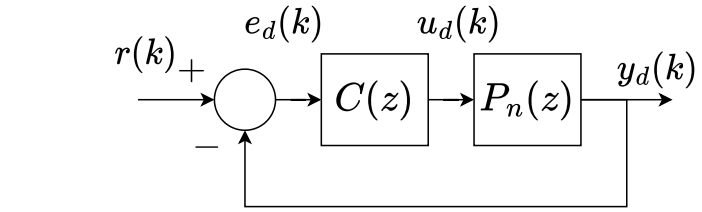
$$e(k) = \frac{1}{1 + C(z)\Delta P(z)}\left(r(k) - n(k)\right). \tag{2.93}$$

Therefore, by augmenting Equation (2.90) with the term $r(k) - r(k)$ and utilizing process data, the design sensitivity function, and the loop gain, and the Equation

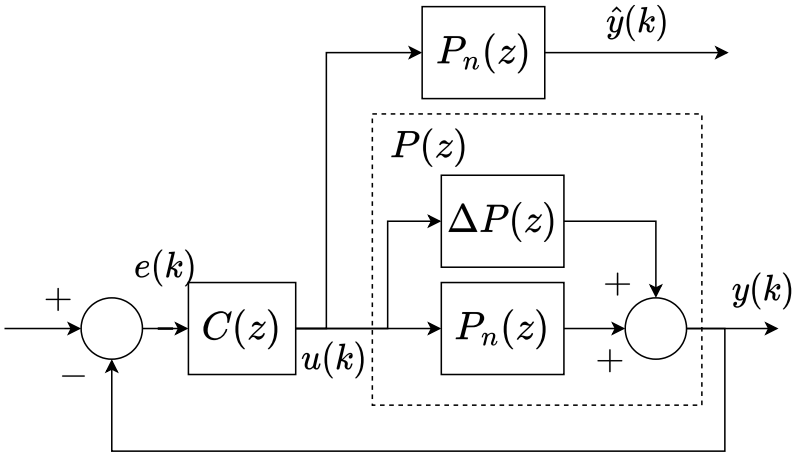
(2.93) the closed-loop design output is expressed as:

$$y_d(k) = r(k) - S_d(z)\,[1 + C(z)\Delta P(z)]\,e(k). \tag{2.94}$$

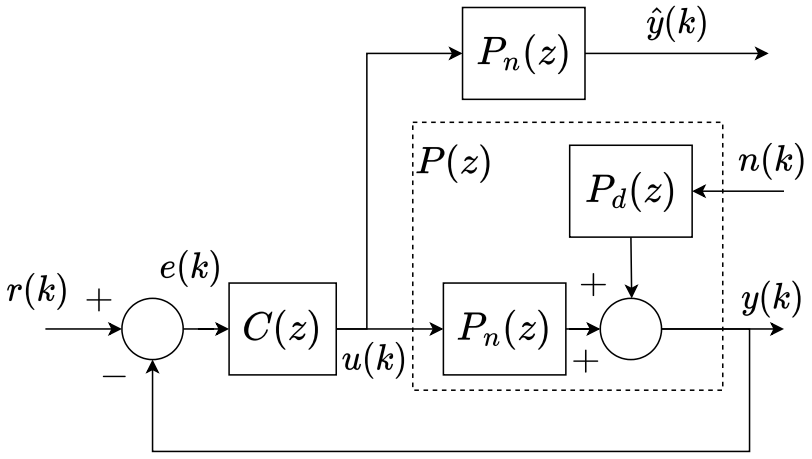
BOTELHO *et al.* (2015, 2016a) suggested an extension of the BADWE *et al.* (2010) work, where application limitations were addressed, especially when working with MPC with soft constraints and avoiding the two data-based model identifications. In these works, the authors proposed the control structure shown in Figure 2.14 for evaluating MPM and UD, where $P_d(z)$ is the disturbance model.



(a) nominal system



(b) with model-plant mismatch



(c) with unmeasured disturbance

Figure 2.14: Schematic diagram of closed-loop.

The designed loop is shown in Figure 2.14a, the system with MPM is shown in Figure 2.14b, and the system with UD is shown in Figure 2.14c. According to these structures, the authors proposed to use the nominal output sensitivity transfer function, $S_o(z)$, which concentrates information on controller tuning, providing the speed of response from each control loop. Therefore, the designed closed-loop output is given by:

$$y_d(k) - y(k) = S_o(z)\left[\hat{y}(k) - y(k)\right], \tag{2.95}$$

where:

$$S_o(z) = \frac{1}{1 + C(z)P_n(z)}. \tag{2.96}$$

Depending on the control structure complexity, $S_o(z)$ can be obtained from the designed loop analytically or by identification using simulation data. From the designed closed-loop output, the authors perform a statistical comparison to differentiate the presence of MPM and UD. In order to distinguish the effect of each case, the nominal output error, $e_0(k) = y(k) - y_d(k)$, was compared with the nominal output, $y_d(k)$ (designed closed-loop output). This comparison shows that if the system has MPM, the reference changes, $r(k)$, correlate $y_d(k)$ and $e_0(k)$. However, when the system is subjected to a UD, $e_0(k)$ depends only on the external signal $n(k)$, while $y_d(k)$ depends only on the reference, $r(k)$.

While the methodologies put forward by BADWE *et al.* (2010) and BOTELHO *et al.* (2015, 2016a) have significantly progressed the field of diagnostic and monitoring methods for control systems, they carry assumptions and limitations that may prevent their application in diverse real-world contexts. Most notably, these approaches rely on the feasibility of achieving an ideal model and easily deriving the projected closed-loop output. This assumption might not always hold in practical scenarios.

In contrast, the present study focuses on the FSP control strategy, a well-established dead-time compensator. The FSP offers more robustness in the face of time delay uncertainties, making it a more suitable choice for time-delay dominant processes commonly encountered in industrial settings.

Additionally, the computational resources required by the methods proposed by Badwe et al. BADWE *et al.* (2010) and Botelho et al. BOTELHO *et al.* (2015, 2016a) might be significant, potentially limiting their effectiveness in real-time applications or systems with computational processing constraints. In response, our work proposes a novel approach where the algorithm can run on a machine parallel to the one implementing the control system, offering significant computational advantages.

However, it is also acknowledged that the entire control framework could be implemented on a single machine. While this might lead to increased computational

costs, the FSP control strategy's robustness and adaptability may outweigh these potential limitations.

In conclusion, these identified limitations of existing methodologies underscore the need for a more pragmatic and robust approach, such as the FSP, better suited to real-world, time-delay dominant processes.

2.6 Final Remarks

This chapter has reviewed the literature concerning model-based predictive control systems, their performance, and the intricacies involved in their tuning. It has also addressed how these systems manage processes with dead time and explored the detection and diagnosis of issues within these systems.

In the literature, various diagnostic techniques have been proposed to enhance the system's dynamic behavior, primarily by updating the model or the controller's parameters and identifying the causes of control performance degradation. The chapter also emphasized the detrimental effects that time delay can introduce on the control structure and the strategies proposed to counter this issue.

Despite the comprehensive body of work on this topic, there is still room for further research, particularly in developing diagnostic strategies for model-based predictive controllers affected by dominant time delay. Moreover, the need for criteria to discriminate between the degradation of the internal model and the onset of unmeasured disturbances is evident.

The subsequent chapters of this thesis introduce and discuss a research methodology to address these gaps. This methodology will focus on improving the tuning process for Model Predictive Control (MPC) systems and diagnosing the Filtered Smith Predictor structure. This focus is grounded on the understanding that these areas present significant challenges in the field and require comprehensive investigation and novel solutions.

Chapter 3

Tuning of Model Predictive Controllers Based on Hybrid Optimization

The first part of this doctoral thesis aims to improve the performance of model-based predictive controllers (MPC) in linear and non-linear systems, emphasizing chemical processes. In this chapter, we present an innovative algorithm for tuning MPC controllers. This approach aims to optimize key tuning parameters, such as the weighting matrices, prediction horizon, and control horizon, using a hybrid combination of two optimization algorithms. The algorithm has been adapted for direct use with MATLAB's Model Predictive Control Toolbox, opening up opportunities for more efficient use and distribution within the scientific community. This work was published in:

- Giraldo, S.A.C., Melo, P.A., & Secchi, A.R. (2022). Tuning of Model Predictive Controllers Based on Hybrid Optimization. *Processes*, 10(2), 351. https://doi.org/10.3390/pr10020351

3.1 Tuning of model predictive controllers based on hybrid-optimization

In the first section of the methodology, a tuning procedure for a model predictive control (MPC) is introduced, specifically designed for multi-input, multi-output systems. This work proposes a tuning method for MPC that is not restricted to a specific prediction model or MPC algorithm. The method is based on the use of two optimization algorithms, forming a hybrid approach: (i) the goal attainment method (GAM) and (ii) a variable neighborhood search (VNS). The motivation of

the present work is to address the problem regarding the nature of the integer variables and the competitive objectives in the formulation of MPC control, without solving a computationally complex mixed-integer dynamic optimization problem.

3.1.1 Goal Attainment Method (GAM)

The operation of a chemical process usually involves the fulfillment of different requirements or specifications in order to reach an optimal point. For instance, low cost, low operating risks, low pollution, high reliability, high quality, and high productivity are reasonable objectives to be achieved in this process (GIRALDO *et al.*, 2022). Nevertheless, most of these goals are often in conflict with each other. Thus, for that reason, a compromise solution must be obtained between them. This is known as a multi-objective optimization problem.

When there are multiple objectives, usually, there is not only a single solution but a set of them, each satisfying one objective to the detriment of the others. This is known as the Pareto set, where the ideal solution is adopted based on the optimization problem's decisions (DEB, 2014). Figure 3.1 shows a geometric representation of the compromise solution, considering a bi-objective problem.

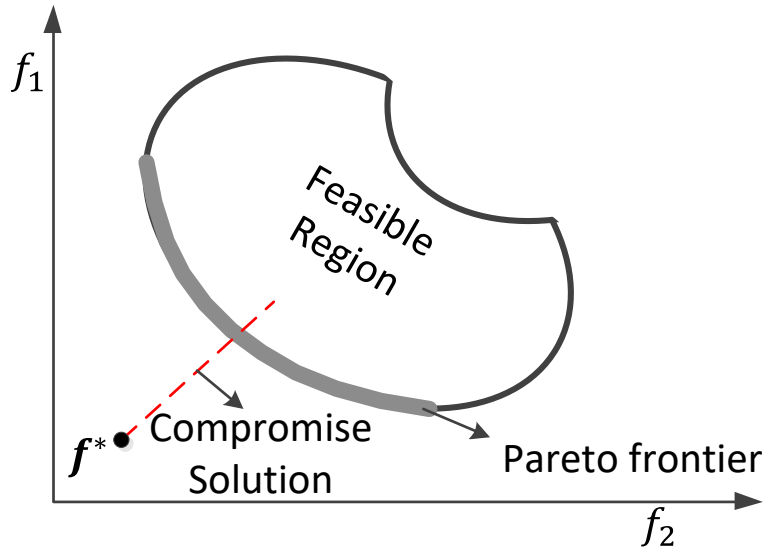


Figure 3.1: Compromise optimization in Pareto frontier for the objective functions f_1 and f_2. f^* is a utopia point.

The major problem in multi-objective optimization can be inferred from Figure 3.1, i.e., the conflict between the objective functions, where the improvement in one of the objective functions may cause the degradation of others. Therefore,

the optimal point in the Pareto curve will depend on a decision for the planned operation of the process.

The GAM formulation, proposed by GEMBICKI (1974), involves expressing a set of utopian goals, $\mathbf{f}^* = [f_1^*, f_2^*, \ldots, f_n^*]$, which is associated with a set of objectives, $\mathbf{f} = [f_1(x), f_2(x), \ldots, f_n(x)]$. $\mathbf{f}^*$ is an unreachable point when minimizing all the objectives simultaneously. The problem's formulation is given by:

$$\begin{aligned} &\min_{\mathbf{x},\gamma} \ \gamma, \\ &\quad \text{subject to} \\ &\quad \mathbf{h}(\mathbf{x}) = 0 \\ &\quad \mathbf{g}(\mathbf{x}) \leq 0 \\ &\quad f_i(\mathbf{x}) - \omega_i \gamma - f_i^* \leq 0, \ i = 1, \cdots, n \\ &\quad \mathbf{L_B} \leq \mathbf{x} \leq \mathbf{U_B}, \end{aligned} \tag{3.1}$$

where $\mathbf{h}(x)$ and $\mathbf{g}(x)$ are equality and inequality constraints, respectively, and ω_i is the relative weight for the i-th objective function $f_i(x)$. $\mathbf{L_B}$ and $\mathbf{U_B}$ are the lower and upper bounds of the decision variables.

The weight vector, $\boldsymbol{\omega}$, measures the relative tradeoffs among the objectives and, in order to eliminate the rigid constraints of the problem, the term $\omega_i \gamma$ is used as a slackness variable.

Equation (3.1) is a convenient way of expressing the commitments among the objective functions, giving greater flexibility to the optimization algorithm to find the best solution. The schematic evolution of this method in the direction of the solution in two dimensions is presented in Figure 3.2 (FLEMING, 1986), where $\mathbf{f}^*$ is the goal and $\mathbf{f_o}$ is the optimal point obtained for a given weight vector, $\boldsymbol{\omega}$, which defines the search direction. The feasible function space, $\Lambda(\gamma)$, shrinks as γ is reduced.

3.1.2 Variable Neighborhood Search (VNS)

The VNS algorithm is a metaheuristic used to solve combinatorial optimization problems, with its operation based on the idea of neighborhood change to find local minima and escape the valleys that contain them (HANSEN *et al.*, 2010). Initially proposed in MLADENOVIĆ and HANSEN (1997), this metaheuristic has been developed in terms of its methods and successfully applied to solve several application problems (ALOISE *et al.*, 2006; HANSEN *et al.*, 2006).

The VNS formulation is given by:

$$\text{minimize } f_v(x) \quad \text{such that } x \in X \text{ and } X \subseteq S, \tag{3.2}$$

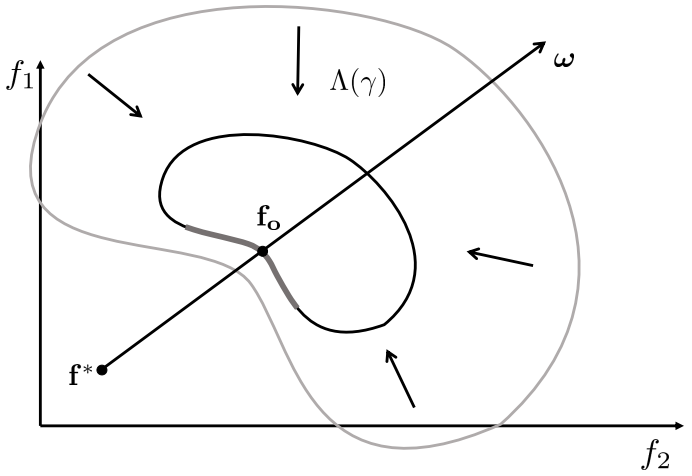


Figure 3.2: Geometrical representation of the goal attainment method.

where S is the solution space, X is the feasible set, x is the feasible solution, and f_v is the real-valued objective function. If S is a finite set, then a combinatorial optimization is established in the VNS algorithm. Otherwise, when $S = \mathbb{R}^n$, the VNS is a continuous optimization. The optimal solution of x^* is given if

$$f_v(x^*) \leq f_v(x), \forall x \in X. \tag{3.3}$$

The VNS algorithm operates as follows. Initially, a starting point x is chosen within the feasible set X. Subsequent to this initialization, the algorithm identifies a direction in which the objective function $f_v(x)$ decreases within a local neighborhood, denoted as $N_e(x)$. Specifically, the "descending direction" is determined by evaluating $f_v(x)$ at different points in $N_e(x)$ and identifying the direction in which $f_v(x)$ exhibits the most significant reduction. The algorithm employs either the *best improvement* or the *first descent* strategies as the criteria for choosing the descending direction. In the former, VNS examines all possible neighborhoods in $N_e(x)$ and selects the one that minimizes $f_v(x)$. In the latter, the algorithm opts for the first neighborhood encountered that yields a decrement in $f_v(x)$. The *first descent* strategy has the potential for rapid improvement if the neighborhoods are searched systematically and the steps likely to produce better results are prioritized. Nonetheless, its worst-case performance equates to that of the *best improvement* strategy, which, while effective in yielding short-term gains, offers no long-term guarantees. Figure 3.3 provides a graphical representation of the VNS methodology.

The VNS algorithm applies random steps in neighborhoods with growing size to diversify the search into combinatorial problems; this is known as shaking (HANSEN *et al.*, 2010). The VNS systematically changes the neighborhood elements to find

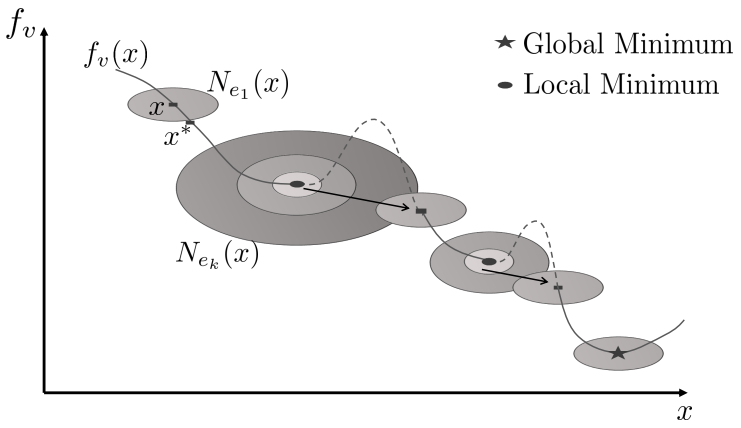


Figure 3.3: The VNS algorithm. Adapted from HANSEN *et al.* (2010).

the optimal one. When the algorithm changes a single neighborhood component, this is known as a *first-order search*. If the algorithm changes two neighborhood components, this is known as a *second-order search* and so on.

3.2 Model Predictive Control Tuning Approach (MPCT)

The MPC controller tuning strategy proposed in this work will be called Model Predictive Control Tuning (MPCT) and is described in this section. Once the MPC is based on a process model, good tuning is also dependent on the quality of this model. In most MPC applications, models are derived by applying experiment design, data collection, and system identification methods. At this stage, the correct selection of the sampling time (T_s) and the scaling of the multivariable model need to be highlighted, as detailed in Appendix A, because these selections strongly affect the tuning performance.

As the MPCT uses a hybrid optimization formulation, it can find the integer and real variables of the tuning problem. The decision variables of the algorithm are $\mathbf{x}_{dv} = [p, \mathbf{m}, \text{diag}(\mathbf{Q}), \text{diag}(\mathbf{W})]$, where integer variables p and $\mathbf{m}$ are determined using the VNS method, and $\mathbf{Q}$ and $\mathbf{W}$ are real-valued diagonal matrices to be found by the GAM algorithm. The lower and upper bounds of the $\mathbf{x}_{dv}$ are $\mathbf{L_B} = [1, \mathbf{1}_{n_u}, \mathbf{1}_{n_y+n_u} \times 10^{-5}]$ and $\mathbf{U_B} = [2^{H_p}, \mathbf{1}_{n_u}(2^{H_c}), \mathbf{1}_{n_y+n_u}(\infty)]$, respectively, where The symbols $\mathbf{1}_{n_u}$ and $\mathbf{1}_{n_y+n_u}$ refer to row vectors of ones of sizes n_u and $n_y + n_u$, respectively, and H_p and H_c are given values such that 2^{H_p} and 2^{H_c} are the upper bounds of p and $\mathbf{m}$.

The MPCT algorithm executes the hybrid optimization (i.e., GAM and VNS

algorithms) sequentially. For this purpose, GAM is the first algorithm to be executed. Minimizing the square error between the closed-loop responses and the pre-established reference trajectories is a common tuning objective reported in diverse MPC tuning procedures (AL-GHAZZAWI *et al.*, 2001; EXADAKTYLOS and TAYLOR, 2010; YAMASHITA *et al.*, 2016). This strategy is employed in the GAM algorithm, where the reference trajectory is set as a tuning parameter. This reference trajectory defines the desirable closed-loop dynamics as a first- or second-order transfer function system with dead-time. Therefore, the multi-objective GAM formulation is given by:

$$f_i(\mathbf{x}_{dv}) = \sum_{k=1}^{\phi} \left[y_i^R(k) - y_i(k, \mathbf{x}_{dv}) \right]^2, i = 1, \cdots, n_y, \tag{3.4}$$

where $y_i^R(k)$ is the discretized reference trajectory of the MIMO output, defined by the user for performance requirements or desirable control behavior; $y_i(k, \mathbf{x}_{dv})$ is the MPC closed-loop trajectory of output i; and ϕ is a predefined tuning horizon, which is large enough to capture the system dynamics.

One may note that the reference trajectory, $\mathbf{y}^R(k)$, is different from the reference tracking signal, $\mathbf{r}(k)$. The reference tracking signal is necessary to obtain the closed-loop trajectory when Equation (2.33) is solved. Due to the MIMO system features, the interactions between the plant variables affect the desired performance for each pre-established variable in the reference trajectory. Therefore, in the GAM algorithm, it is recommended to establish a similar reference tracking signal to that established in the real plant in both the reference trajectory and the closed-loop trajectory because the algorithm minimizes the effect of the MIMO system interactions by minimizing Equation (3.4).

The utopian solution was initially proposed in GIRALDO *et al.* (2019) by solving the optimization problems defined as:

$$\begin{aligned} f_i^*(\mathbf{x}_{dv}) = & \min_{\mathbf{x}_{dv}} f_i(\mathbf{x}_{dv}), \ i = 1, \cdots, n_y \\ & \text{subject to} \\ & \mathbf{L_B} \leq \mathbf{x}_{dv} \leq \mathbf{U_B}. \end{aligned} \tag{3.5}$$

Nevertheless, a utopian solution is infeasible because not all the tuning objectives have the same optimal point. We propose removing this optimization stage to improve the computational cost of the GAM algorithm. In this case, a reasonable selection of the utopia point is to select it as zero because the set of objectives, shown in Equation (3.4), are positive functions and because it meets the criteria of being a utopia point. Equation (3.4) can only be zero when each objective follows

the reference trajectory perfectly. Thus, the optimization problem of the GAM algorithm, shown in Equation (3.1), is solved using sequential quadratic programming to find the tuning parameters. Once GAM predetermines the weight matrices, the MPCT searches the prediction and control horizons using VNS.

p and $\mathbf{m}$ are integer parameters of the objective function J, Equation (2.33). These parameters are converted into binary numbers of which the maximum sizes in bits are H_p and H_c, respectively; for instance, if p is a $H_p = 4$-bit variable, then it has a maximum prediction horizon size of 15.

The MPCT finds only one prediction horizon for the system and different control horizons for each manipulated variable. The algorithm takes the slowest dynamics to define the prediction horizon in this context. If a linear system transformation is made and then diagonalized, slow eigenvalues dominate the system's dynamic modes. Moreover, since the process is a MIMO system with interactions between variables, the algorithm needs to simulate the system until the slowest dynamics are captured; therefore, only one prediction horizon is defined. For each manipulated variable, it is possible to define a different control horizon to manipulate the process, minimize the computational cost, and improve the system's dynamic response.

3.2.1 Obtaining Optimal Horizons

The selection of the control and prediction horizons within an MPC strategy depends largely on the system's dynamics to be controlled; that is, it depends on whether the system is stable, unstable, oscillatory, non-minimal phase, etc. Intuitively, it can be noted that the design of an MPC controller in an unconstrained way must show a good performance. Therefore, if the desired performance is not achieved, this controller tuning will most likely not control the system with the active constraints.

To obtain a good selection of the MPC horizons, one should ask what is a well-posed optimization problem. To answer this question, one may start from the fundamental concept of an MPC controller, where the controller uses the system model (linear or nonlinear) to solve the objective function (with or without constraints) shown in Equation (2.33). In this context, the MPC, in a given Pareto front, finds an optimal trajectory (open loop) to be applied in the control law. However, as stated before, only the first action is applied to the process because there will be updated process measures in the next sampling time that allow the trajectory correction, solving Equation (2.33) again. This is known as the receding horizon (CAMACHO and BORDONS, 2002). However, if one only considers this first calculation, in $k = 1$, and if the prediction and control horizons are poorly selected, the prediction of this first trajectory will differ significantly from the closed-loop behavior of the system when the receding horizon concept is applied. The predictions do not make

sense because the optimization does not represent what will happen in the future.

Optimizing the objective function J, as shown in Equation (2.33), will be especially useful if the trajectories are close to the response of the closed-loop system, both of which are calculated with the internal model of the controller. For this, the internal model of the MPC is taken, with the VNS algorithm establishing a reference, $\mathbf{r}_s(k)$, for the internal model. Finally, the controller algorithm is solved. With this solution, it is possible to obtain the output response and the control action in a closed loop, $\mathbf{y}(k)$ and $\mathbf{u}(k)$, respectively, using the receding horizon concept and remembering that $\mathbf{u}(k) = \Delta\mathbf{u}(k) + \mathbf{u}(k-1)$. However, it is also possible to calculate the trajectories of the outputs and inputs of the system (open-loop responses) $\mathbf{y}_o(k|1)$ and $\mathbf{u}_o(k|1)$, respectively, which is the first optimization that the MPC made at the first sampling time.

3.2.2 Variation on the Prediction and Control Horizon

To illustrate the influence of the horizons on the closed-loop dynamics of the MPC controller, 4 SISO systems were chosen to perform a preliminary analysis.

Stable system:

$$G_1(z^{-1}) = \frac{z^{-1} + 0.3z^{-2}}{1 - 1.2z^{-1} + 0.32z^{-2}}. \tag{3.6}$$

Non-minimal phase system:

$$G_2(z^{-1}) = \frac{z^{-1} - 2z^{-2}}{1 - 1.7z^{-1} + 0.72z^{-2}}. \tag{3.7}$$

Unstable system:

$$G_3(z^{-1}) = \frac{z^{-1} + 0.4z^{-2}}{1 - 1.7z^{-1} + 0.6z^{-2}}. \tag{3.8}$$

Oscillatory system:

$$G_4(z^{-1}) = \frac{z^{-1} + 0.4z^{-2}}{1 - 0.9z^{-1} + 0.6z^{-2}}. \tag{3.9}$$

Intuitively, it can be noticed that the design of an MPC controller in an unconstrained way must show a good performance. Therefore, if the desired performance is not achieved, this controller tuning will most likely be unable to control the system with the active constraints.

The first tests on these systems considered a short prediction horizon with a control horizon equal to 1 and weight matrices equal to the identity matrix. Figure 3.4 shows the response of the systems to the variation of the prediction horizon.

This figure compares the closed-loop behavior of the MPC controller, $[\mathbf{y}(k),\ \mathbf{u}(k)]$, with a receding horizon, with the trajectory calculated at the first sampling time, $[\mathbf{y_o}(k|1),\ \mathbf{u_o}(k|1)]$. One may note that $\mathbf{u_o}(k|1)$ is an $n_u \times m$ vector, so the last value is repeated to complete the simulation time, as one can see in the projection of $\mathbf{u_o}(k|1)$. In this specific scenario, due to $m = 1$, there is only a singular control action.

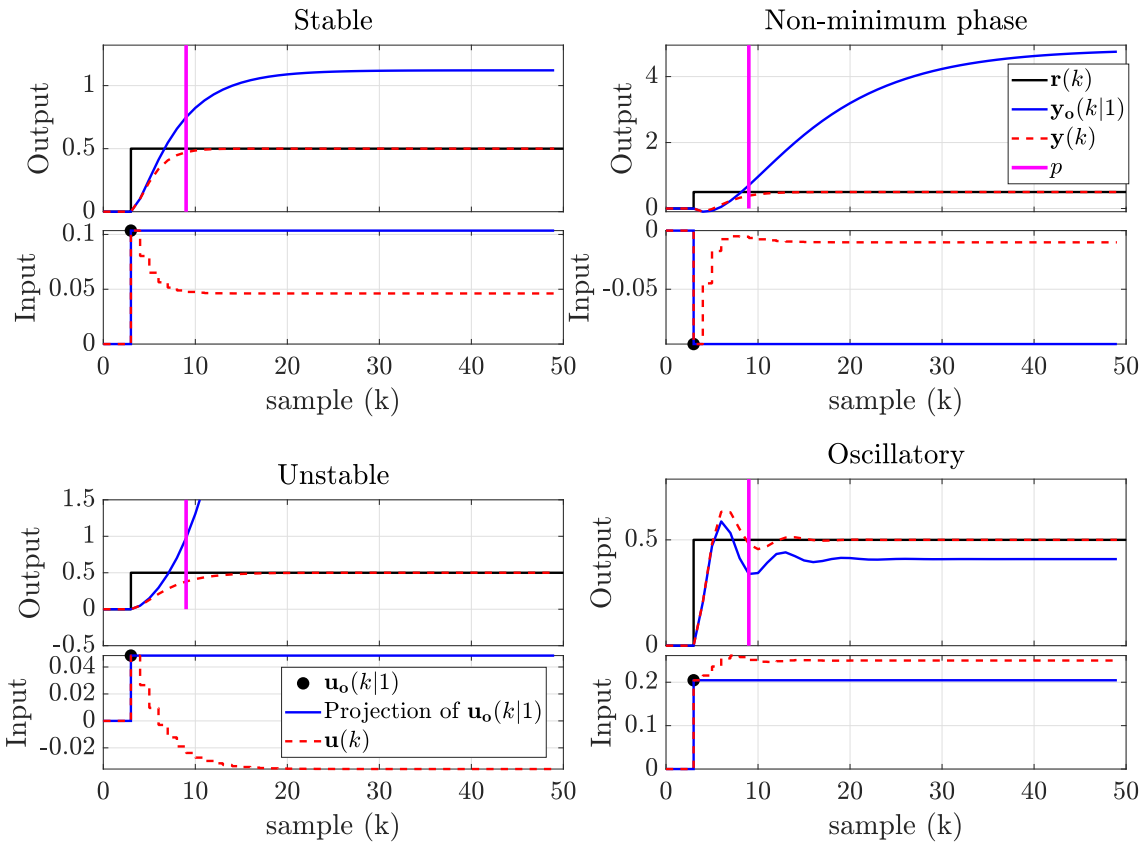


Figure 3.4: Optimized prediction vs closed-loop behavior with $p = 5$ and $m = 1$.

As shown in Figure 3.4, the systems' predictions appear adequate within the specified prediction horizon, spanning from the step change at $k = 4$ until the end of the prediction horizon at $k = 9$. However, across all four scenarios, the predictive strategy is generally weak. The discrepancy between the closed-loop and open-loop behaviors is evident, even within the prediction horizons, indicating an ill-structured optimization. Any effective performance of the closed-loop system appears more coincidental than a result of deliberate design. This suggests the system's initial forecast to reach the reference is flawed. With each sampling period, the system is forced to alter the previously taken direction and decision.

This pattern reveals that the performance index J only penalizes the tracking errors within p samples without considering the implicit follow-up after that point.

Figure 3.5 presents the systems' behavior when the prediction horizon is ex-

tended to $p = 15$. The prediction appears to mirror its open-loop behavior in this scenario more closely. It can be observed that the prediction of the steady-state error diminishes as the prediction horizon increases, indicating a reduced impact on future terms.

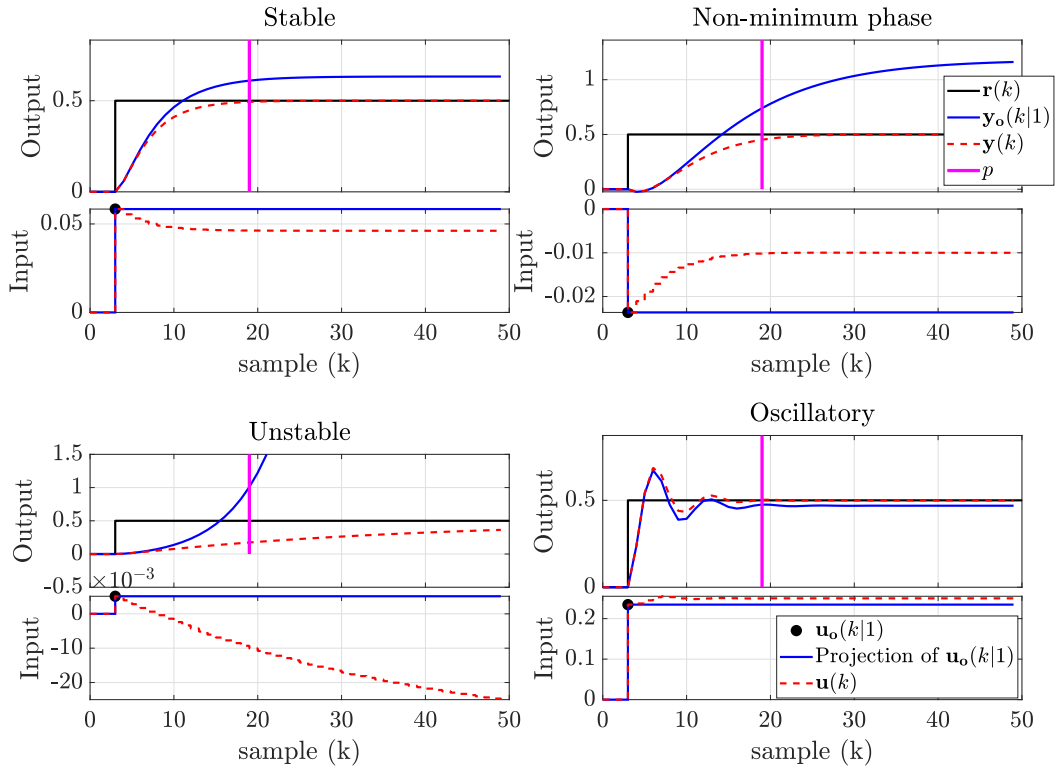


Figure 3.5: Optimized prediction vs closed-loop behavior with $p = 15$ and $m = 1$.

This tuning yields close optimal control movements to the desired steady-state in stable, oscillatory, and non-minimal phase processes. The closed-loop behavior closely mirrors the dynamics of the open-loop. However, a single control action in unstable processes leads to a divergent prediction. As a result, achieving a prediction close to the goal becomes infeasible, rendering any optimization ill-posed. Thus, selecting a large p is advantageous, except in unstable systems, where the risk lies in steady-state errors, potentially saturating transient errors.

Figure 3.6 showcases the effect of increasing the control horizon to $m = 4$. It becomes clear that a higher m value enhances the prediction, and if the prediction horizon is large, the steady-state error is minimal.

Nevertheless, while a larger control horizon generally improves prediction, the same cannot be said for situations where the prediction horizon is small, as can observed in Figure 3.7. In such cases, the prediction often lacks a clear direction, making selecting these horizons unwise, especially in non-minimum phase and unstable systems.

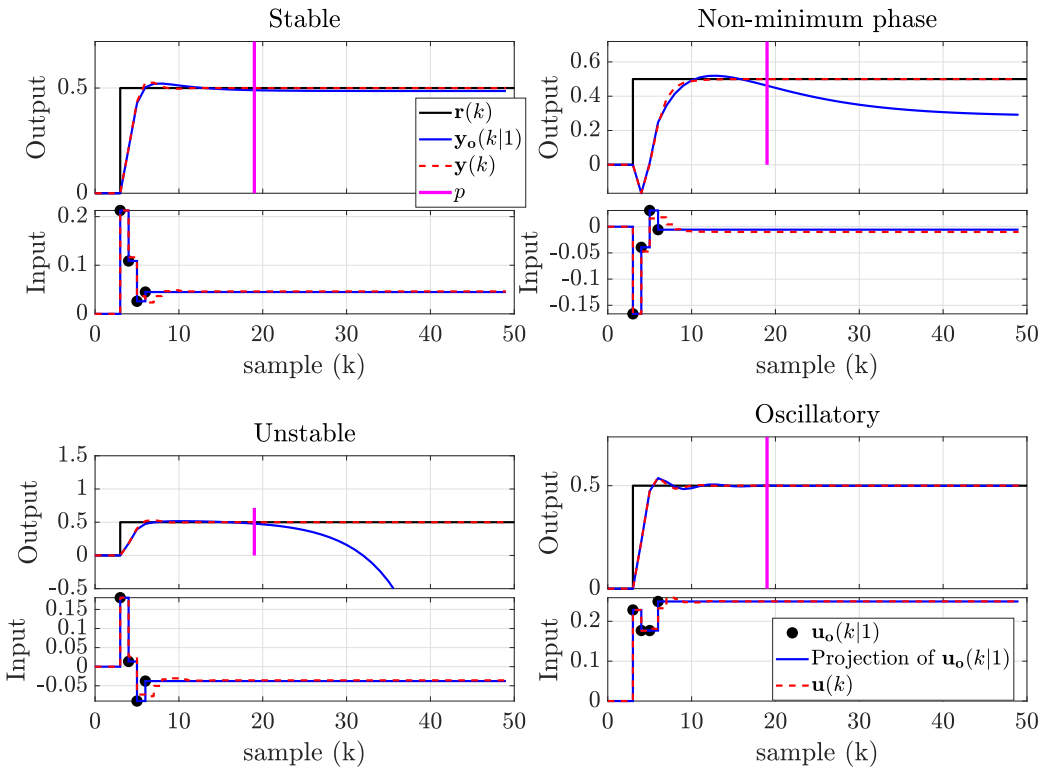


Figure 3.6: Optimized prediction vs closed-loop behavior with $p = 15$ and $m = 4$.

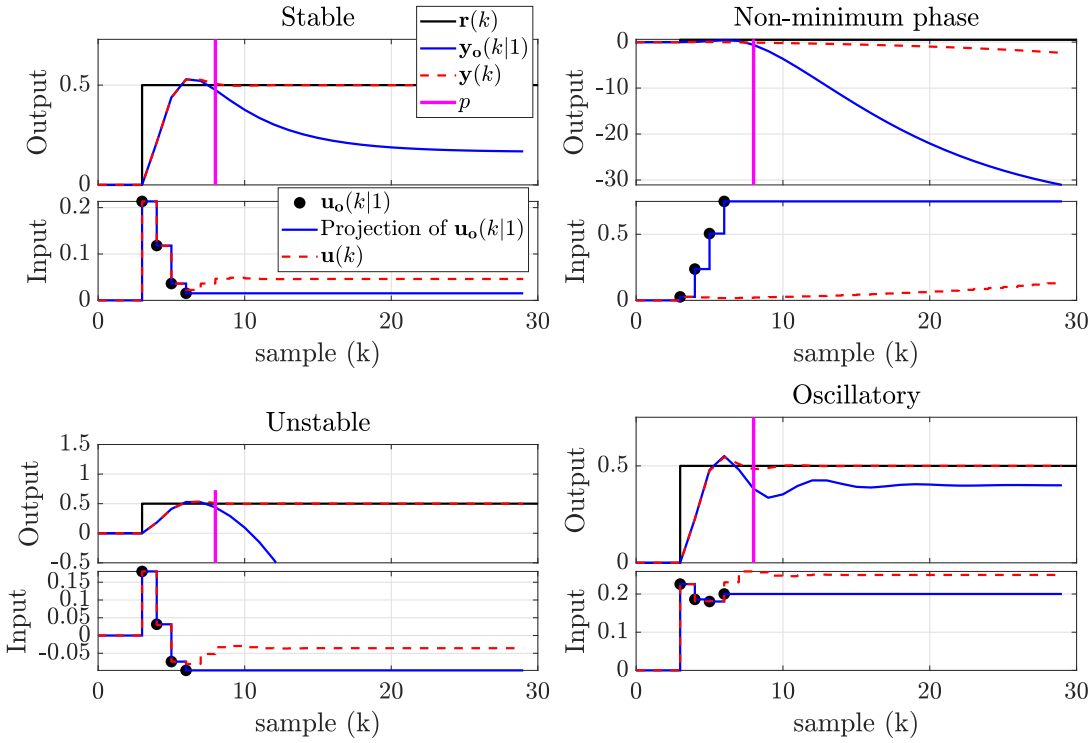


Figure 3.7: Optimized prediction vs closed-loop behavior with $p = 4$ and $m = 4$.

Figures 3.6 and 3.8 show that larger p and m values can improve prediction. This usually leads to good results both within and beyond the prediction horizon, except in unstable systems. However, making m too large may not bring further benefits, as the control steps $\mathbf{u}_o(k|1)$ become more and more similar. This causes only minor changes to the manipulated variable $\Delta\mathbf{u}(k)$, providing little additional information and increasing the computational load for the MPC. As depicted in Figure 3.8, an overly large control horizon m makes the optimization algorithm more computationally demanding.

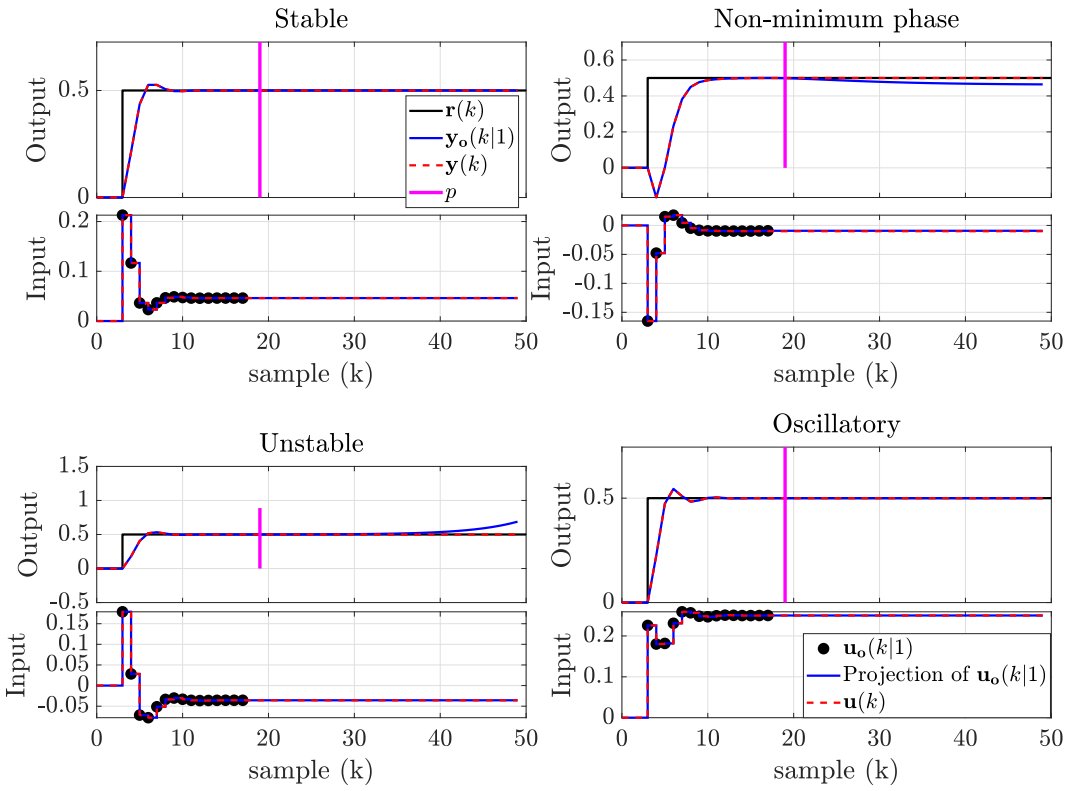


Figure 3.8: Optimized prediction vs closed-loop behavior with $p = 15$ and $m = 15$.

In conclusion, it is imperative to properly tune sufficiently large prediction and control horizons to ensure a well-posed optimization. The prediction horizon should encapsulate adequate information about steady-state errors, allowing long-term predictions to adhere to the desired trajectory. Evidence of this can be seen as predictions approach the steady-state. Meanwhile, the control horizon ought to incorporate enough terms to: (i) adjust the stationary state, (ii) counteract any external dynamics from the open loop, and (iii) maintain flexibility in optimizing transient performance.

3.2.3 Tuning Prediction and Control Horizons via VNS Algorithm

The VNS algorithm must find the appropriate sizes for the prediction and control horizons to match both trajectories. The algorithm uses the *first-descent* direction with a *first-order search*, that is, modifying a single bit from the initial condition and solving Equation (3.10). When the algorithm does not find other solutions, a *second-order* and then *third-order search* are applied to escape from local minima. If the *third-order search* is executed and the algorithm does not find another solution, then the VNS method selects the horizons with the lowest cost found in its search.

$$\begin{aligned}\min_{p,\mathbf{m}} \Bigg\{ f_v(\mathbf{x}_{dv}) = & \sum_{i=1}^{n_y} \left[\sum_{k=1}^{\phi} \left(\left\{ y_i(k) - y_{o_i}(k|1) \right\}^2 + \left\{ y_i^R(k) - y_i(k) \right\}^2 \right) \right] \\ & + p + \sum_{j=1}^{n_u} \left[\sum_{k=1}^{m_j} \frac{|u_{o_j}(1|1)|}{|u_{o_j}(k+1|1) - u_{o_j}(k|1)|} \right]^2 \Bigg\} \\ & \text{subject to} \\ & \text{Equations (2.33b) to (2.33g)} \\ & u_{o_j}(k+1|1) \neq u_{o_j}(k|1) \\ & \mathbf{m} < p,\end{aligned} \tag{3.10}$$

where the first term of the objective function seeks to minimize the distance between the closed-loop, $y_i(k)$, and the trajectory calculated at the first sampling time, $y_{oi}(k|1)$, of the output i. This term is used to establish the performance criteria desired by the user. It can be used to ensure the robustness of the controller against model uncertainties by establishing more conservative dynamics. The second term seeks to minimize the distance between the reference trajectory, $y_i^R(k)$, and the closed-loop response, $y_i(k)$, and the last two terms of the function avoid the selection of large p and $\mathbf{m}$, respectively. One may note that this method avoids selecting a large control horizon in the last term as long as the rate of change in the denominator is significant. For instance, if the rate of change is close to zero, this term is penalized.

The VNS algorithm is presented in Algorithm 1 where $\dot{\mathbf{x}}$ is the process model (state-space representation), and $\dot{\mathbf{x}}^R$ is the state-space model for the reference trajectory. The algorithm optimizes two primary parameters in the provided pseudocode: prediction and control horizons. Initiating with a first-order search, the algorithm modifies specific bits of these parameters sequentially. For each order of investigation, both p and $\mathbf{m}$ are transformed into their respective binary representations. Subsequently, distinct bits are altered based on the current order to generate new configurations. Each such configuration undergoes evaluation against the objective

function presented in Equation (3.10). If a configuration proves optimal relative to the function, it is preserved. This systematic approach facilitates a thorough exploration of potential configurations, seeking the most efficacious arrangement for both p and $\mathbf{m}$.

Algorithm 1: VNS algorithm

Input : $\dot{\mathbf{x}}$, $\dot{\mathbf{x}}^R$, $\mathbf{r}_s$, p^0, $\mathbf{m}^0$, $\mathbf{Q}^0$, $\mathbf{W}^0$, ϕ, H_p, H_c, f_v^a
Output: p, $\mathbf{m}$, f_v

$order \leftarrow 1$; // first-order search
while $order \leq 3$ **do**
  $tt \leftarrow 1$; // $tt = 1$ varies p, $tt = 2$ varies $\mathbf{m}$
  while $tt \leq 2$ **do**
    if $tt == 1$ **then**
      $X_1 \leftarrow bits(p^0)$; // convert p^0 to bits
      $k_{max} \leftarrow H_p$; // max. p neighborhood
    else
      $X_1 \leftarrow bits(\mathbf{m}^0)$; // convert $\mathbf{m}^0$ to bits
      $k_{max} \leftarrow H_c$; // max. $\mathbf{m}$ neighborhood
    end
    $k \leftarrow 1$;
    while $k \leq k_{max}$ **do**
      // varies "order" bits
      $X_2 \leftarrow \text{NeighborhoodChange}(X_1, k, order)$;
      // evaluate Equation (3.10)
      $f_v \leftarrow \text{Obj\_Function}(X_2, \mathbf{r}_s, \dot{\mathbf{x}}, \dot{\mathbf{x}}^R, \mathbf{Q}^0, \mathbf{W}^0, \phi, H_p, H_c)$;
      if $f_v \leq f_v^a$ **then**
        $f_v^a \leftarrow f_v$; $X_1 \leftarrow X_2$; $k \leftarrow 1$;
        [p, $\mathbf{m}$]$\leftarrow$SaveSolution(X_1,tt);
      else
        $k \leftarrow k + 1$;
      end
    end
    $tt \leftarrow tt + 1$;
  end
  $order \leftarrow order + 1$;
end

3.2.4 Variation on the Weight Matrices

This section evaluates how variations in the weight matrices $\mathbf{Q}$ and $\mathbf{W}$, adjusted by the GAM algorithm, impact the dynamics and control responses of the discussed systems. Initially, it is considered the prediction horizon $p = 15$ and the control horizon $m = 4$. The weight matrix $\mathbf{W}$ is increased, which is responsible for weighing the increments of control of the MPC, while the weighting matrix $\mathbf{Q}$ is set equal to

the identity matrix. This result can be observed in Figure 3.9.

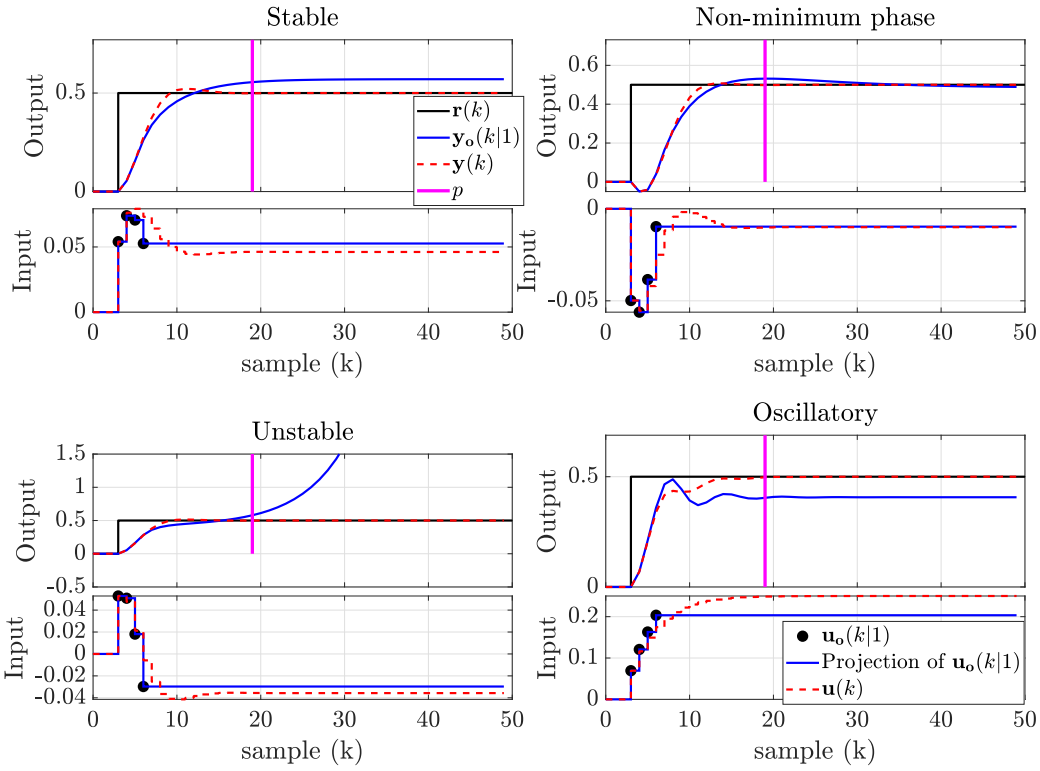


Figure 3.9: Optimized prediction vs closed-loop behavior with $\mathbf{Q} = \mathbf{I}$ and $\mathbf{W} = 50\mathbf{I}$.

Figure 3.9 illustrates that a larger $\mathbf{W}$ results in smaller control increments and a slower tracking of the reference trajectory. The predictions outside the horizon are suboptimal, suggesting that optimizing the J index may not be appropriate.

In Figure 3.10, the weight matrix $\mathbf{Q}$ is increased, and the matrix $\mathbf{W}$ is set to the identity matrix. As the $\mathbf{Q}$ weighs the reference tracking, it is clear that the control actions on the system are more aggressive, which may consume more energy, and tend to minimize the error in the steady-state. This causes an acceleration in the transient response, and the predictions seem to improve inside and outside the prediction horizon. In this case, the prediction is close to the closed-loop dynamics, indicating a well-posed optimization.

In light of the earlier discussion on the MPCT approach, the results obtained with varying weight matrices $\mathbf{Q}$ and $\mathbf{W}$ underline the importance of these parameters in the MPC optimization process. In particular, the adjustment of these matrices directly impacts the controller's ability to minimize the error between the system response and a user-defined reference trajectory, as expressed in Equation (3.4). As shown, larger $\mathbf{Q}$ and $\mathbf{W}$ values yield distinct control behavior and prediction performance, emphasizing the need for a well-defined tuning strategy such as MPCT. The implications of these behaviors are tackled by the GAM algorithm, where the

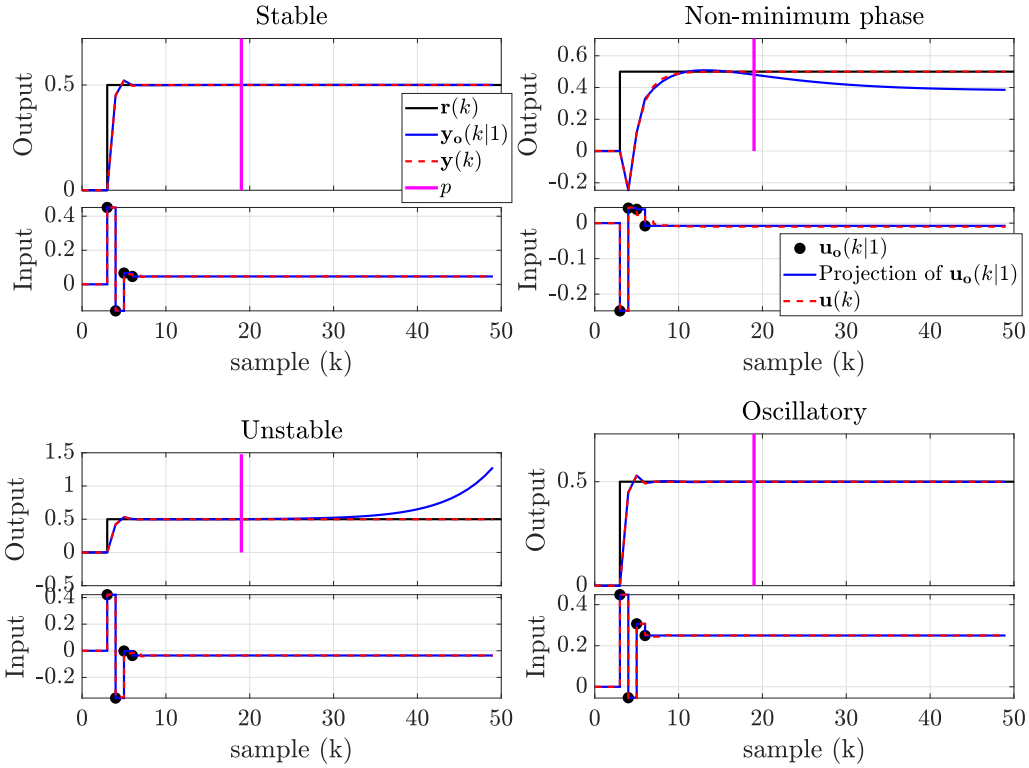


Figure 3.10: Optimized prediction vs closed-loop behavior with $\mathbf{Q} = 50\mathbf{I}$ and $\mathbf{W} = \mathbf{I}$.

optimal weight matrices are determined. This interplay between parameter tuning and system performance underscores the necessity of comprehensive optimization approaches in achieving desired control outcomes.

Robust Stability Analysis

An important point about tuning methods is performance/robustness requirements. Heuristic methods generally have pre-set adjustment criteria, whereas self-tuning methods are based on the desired response information (FONTES *et al.*, 2019). Since two optimizers are working together, as shown in Figure 3.11, robustness is not an exclusive task of the tuning algorithm. The MPCT works in conjunction with the MPC controller, where the primarily responsible for dealing with the problem's robustness is the optimizer of the MPC controller. In this context, the robustness project must be considered in the MPC formulation if fast tuning is required. However, to meet either the robustness or the performance criteria on the MPCT, the robustness in the tuning method will be explicitly considered.

To ensure a robust method, the tuning parameters in this work can be adjusted in different ways since GAM and VNS algorithms use the internal model of the MPC: (i) it is possible to use the same robust formulation of the MPC in the MPCT to find the tuning parameters, e.g., through minimax optimization formulations in

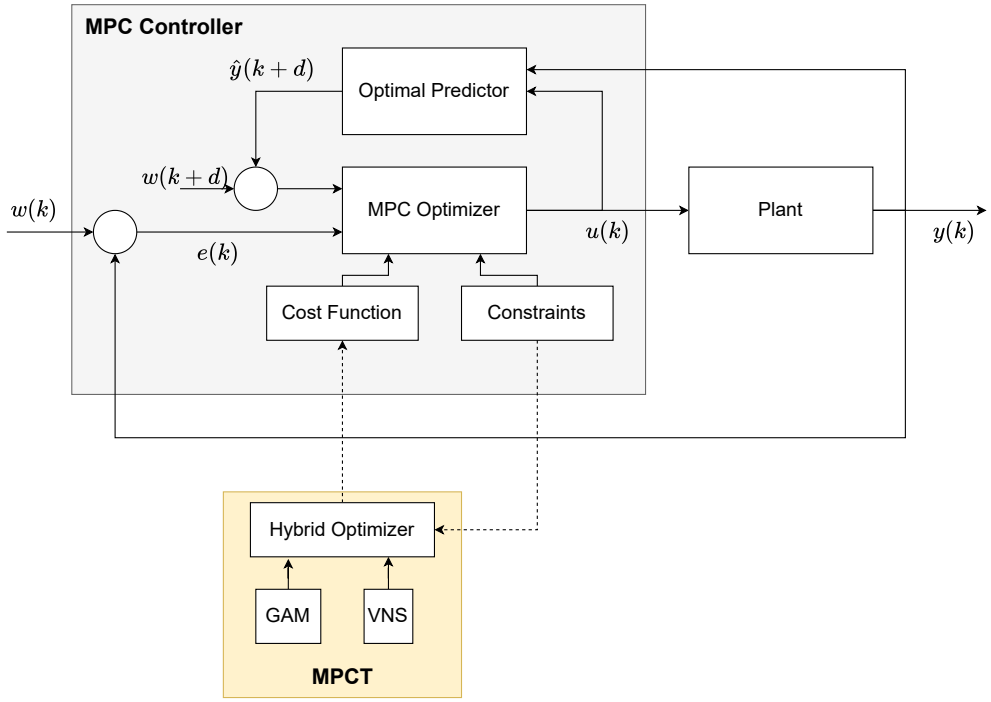


Figure 3.11: MPC + MPCT structure.

both algorithms; (ii) it is possible to estimate the tuning parameters under a worst-case control problem using a family of models that represent the dynamic behavior of the process, and finally; (iii) robustness can be considered through the result of a conservative performance since the MPCT algorithm employs a reference trajectory defined by the user.

Problem constraints are also considered by the tuning algorithm, limiting the search region of the controller parameters. In general terms, the same optimizer used in the controller is employed within the optimization of the MPC tuning algorithm, provided that the optimization problems are similar. Should the optimization problems of the MPC and MPCT differ in nature, as could occur if one is quadratic and the other non-linear, appropriate adaptations will be required to ensure robustness against modeling errors.

3.2.5 MPCT Algorithm

The Model Predictive Control Tuning (MPCT) implementation is presented in the Algorithm 2. The initialization phase encompasses the scaling of the system model, resulting in $\dot{\mathbf{x}}_e$, and the setting of initial cost values to a high magnitude to facilitate improved approximations in subsequent iterations. Central to the algorithm are two intertwined sub-algorithms: the GAM and the VNS. In each cycle, the GAM algorithm optimizes the weight matrices $\mathbf{Q}$ and $\mathbf{W}$, considering the objective function presented in Equation (3.4). Subsequently, the VNS algorithm refines the predic-

tion and control horizons while incorporating the updated weight matrices. The overarching MPCT algorithm undergoes iterative loops until a pre-defined stopping criterion, predicated on the iteration count surpassing a threshold 'Stop', is satisfied. The culminating output consists of the optimized values for the prediction horizon, control horizon, and the weight matrices $\mathbf{Q}$ and $\mathbf{W}$.

Algorithm 2: MPCT algorithm

Input : $\dot{\mathbf{x}}$, $\dot{\mathbf{x}}^R$, $\mathbf{r}$, $\mathbf{r}_s$, $\boldsymbol{\omega}$, ϕ, $\mathbf{x}_{0_{dv}}$, H_p, H_c, $Stop$
Output: p, $\mathbf{m}$, $\mathbf{Q}$, $\mathbf{W}$

$[p^0, \mathbf{m}^0, \mathbf{Q}^0, \mathbf{W}^0] \leftarrow \mathbf{x}_{0_{dv}}$; `// Initialization`
$\dot{\mathbf{x}}_e \leftarrow \text{scale}(\dot{\mathbf{x}})$; `// scale the system model`
$f_v^a \leftarrow 10^8$; $F_T \leftarrow 10^8$; `// Initialization of the costs`
$k \leftarrow 1$; $Ite \leftarrow 0$;
while $k == 1$ **do**
  $[\mathbf{Q}, \mathbf{W}, f_g] \leftarrow \text{GAM}(\dot{\mathbf{x}}_e, \mathbf{r}, p^0, \mathbf{m}^0, \mathbf{Q}^0, \mathbf{W}^0, \phi)$; `// GAM Algorithm`
  $[\mathbf{Q}^0, \mathbf{W}^0] \leftarrow [\mathbf{Q}, \mathbf{W}]$;
  `// VNS Algorithm`
  $[p, \mathbf{m}, f_v] \leftarrow \text{VNS}(\dot{\mathbf{x}}_e, \dot{\mathbf{x}}^R, \mathbf{r}_s, p^0, \mathbf{m}^0, \mathbf{Q}^0, \mathbf{W}^0, \phi, H_p, H_c, f_v^a)$
  $[p^0, \mathbf{m}^0, f_v^a] \leftarrow [p, \mathbf{m}, f_v]$;
  if $f_g \leq F_T$ **then**
    $F_T \leftarrow f_g$;
  end
  `// Stopping criterion`
  if $Ite > Stop$ **then**
    $k \leftarrow 2$;
  end
  $Ite \leftarrow Ite + 1$;
end

As an added contribution, the source code for the MPCT algorithm is made readily available for anyone interested in utilizing this methodology. Specifically, a MATLAB adaptation is provided, facilitating seamless implementation alongside the MATLAB MPC Toolbox. Notably, this MATLAB version presents a specific approach to the proposed method, where selecting multiple control horizons $\mathbf{m}$ is not possible. Importantly, this simplification does not compromise the effectiveness of the proposed solution; instead, it eases the problem's complexity while maintaining the fundamental essence of the MPCT approach.

A Matlab routine *fgoalattain* was used for the GAM optimization problem. The termination tolerance for the function value, the constraint violation, and the first-order optimality were set to 10^{-6}. The problems described here were solved using an Intel®Core i7 8750H 2.2GHz, 16 GB RAM computer.

The MATLAB adaptation of the MPCT algorithm can be accessed directly from the following GitHub repository: https://github.com/sergioacg/Model-Predictive-

Control/tree/main/MPC-Tuning.

3.3 Simulation Case Studies

This section presents a series of case studies conducted using a range of benchmark processes from the literature. These cases will provide an in-depth exploration of the Model Predictive Control Tuning (MPCT) methodology, demonstrating its implementation, performance, computational cost, and adaptability to various scenarios. By employing different MPC formulations and applying the MPCT to linear, nonlinear, square, and non-square systems, the aim is to offer a comprehensive understanding of its versatility and effectiveness.

Furthermore, as an added resource, the source code for the simulations of these case studies, as well as the MPCT algorithm itself, is made readily available in the following GitHub repository: https://github.com/sergioacg/Model-Predictive-Control/tree/main/MPC-Tuning. It should be noted that the simulations in the repository utilize the specific MATLAB adaptation of the MPCT. This particular adaptation is designed for direct implementation alongside the MATLAB MPC Toolbox, which only accepts a single control horizon, yet this does not compromise the effectiveness of the proposed solution.

3.3.1 The Subsystem of the Shell Heavy Oil Fractionator — A Square MIMO System with Linear MPC Formulation

A 3×3 MIMO subsystem of the Shell Heavy Oil Fractionator (HOF) benchmark system presented in MACIEJOWSKI (2002) was tuned to demonstrate the performance of the MPCT algorithm. For this case study, a linear MPC formulation known as generalized predictive control (GPC) was selected. It should be noted that while other linear formulations are also applicable for this problem, stability guarantees, particularly in the absence of terminal ingredients, should be carefully addressed and verified.

The three inputs of the system, u_1, u_2, and u_3, are the top draw flow rate, the side draw flow rate, and the bottom reboiler heat duty, respectively. The three controlled outputs, y_1, y_2, and y_3, are the top end point composition, the side end point composition, and the bottom reboiler temperature, respectively. This system is represented by the following transfer functions:

$$\mathbf{P_n}(s) = \begin{bmatrix} \dfrac{4.05+2.11\epsilon_1}{50s+1}e^{-27s} & \dfrac{1.77+0.39\epsilon_2}{60s+1}e^{-28s} & \dfrac{5.88+0.59\epsilon_3}{50s+1}e^{-27s} \\ \dfrac{5.396+3.29\epsilon_1}{50s+1}e^{-18s} & \dfrac{5.72+0.57\epsilon_2}{60s+1}e^{-14s} & \dfrac{6.90+0.89\epsilon_3}{40s+1}e^{-15s} \\ \dfrac{4.38+3.11\epsilon_1}{33s+1}e^{-20s} & \dfrac{4.42+0.73\epsilon_2}{44s+1}e^{-22s} & \dfrac{7.20+1.33\epsilon_3}{19s+1} \end{bmatrix}, \quad (3.11)$$

where ϵ_i are the uncertainties in the gain model and the time constants are given in minutes.

The first step of the algorithm is to scale the process model using the following diagonal matrices: $\mathbf{L} = \text{diag}[0.617; 0.595; 0.840]$ and $\mathbf{R} = \text{diag}[1.0; 0.416; 0.622]$, which are found through the solution of Equation (A.1). The scaled gain matrix for the nominal case is given by:

$$\mathbf{K} = \begin{bmatrix} 2.4983 & 0.4546 & 2.2563 \\ 3.2087 & 1.4179 & 2.5552 \\ 3.6784 & 1.5457 & 3.7614 \end{bmatrix}. \quad (3.12)$$

A zero-order hold discretization of the scaled process transfer function is used with a $T_s = 4$ min sampling period.

The MPCT parameters are set as: $p = 255$ (8-bits), $\mathbf{m} = [15, 15, 15]$ (4-bits each), and $\phi = 400$.

Two reference trajectories for the HOF are established to demonstrate the desired performance and robustness of the MPC tuning parameters. Therefore, the two reference trajectories are commanded by first-order plus dead-time systems with static gain $\mathbf{k}^R = [1.0, 1.0, 1.0]$ and dead-time $\mathbf{L}^R = [27.0, 14.0, 0.0]$. The difference between the two references is in the time constant. The first one, named case 1, has an aggressive dynamics with the time constant $\boldsymbol{\tau_a}^R = [5.0, 9.0, 5.7]$. The second one, named case 2, has a conservative dynamics with the time constant $\boldsymbol{\tau_c}^R = [30.0, 30.0, 30.0]$. The relative weight for GAM was set as $\boldsymbol{\omega} = [0.40, 0.05, 0.55]$.

Table 4.1 presents the solution of the MPCT tuning procedure for the two scenarios.

The required computational time was 35 min for case 1 and 40 min for case 2. Tuning the MPC controller for the HOF benchmark was reported in VALLERIO *et al.* (2014); YAMASHITA *et al.* (2016), in which the computational time required for the *lexicographic tuning technique* optimization method was 4.27 h, for the *compromise tuning technique* it was 53 min, and for the *normal boundary intersection* it was 20.1 h.

Figures 3.12 - 3.14 show the resulting trajectories for the MPCT, where one may notice that the VNS algorithm estimated an adequate size for prediction and control horizons since the closed-loop response, $\mathbf{y}(k)$, is as close as possible to the output trajectory at the first optimization, $\mathbf{y}_o(k|1)$, in both scenarios obtaining small values for both horizons.

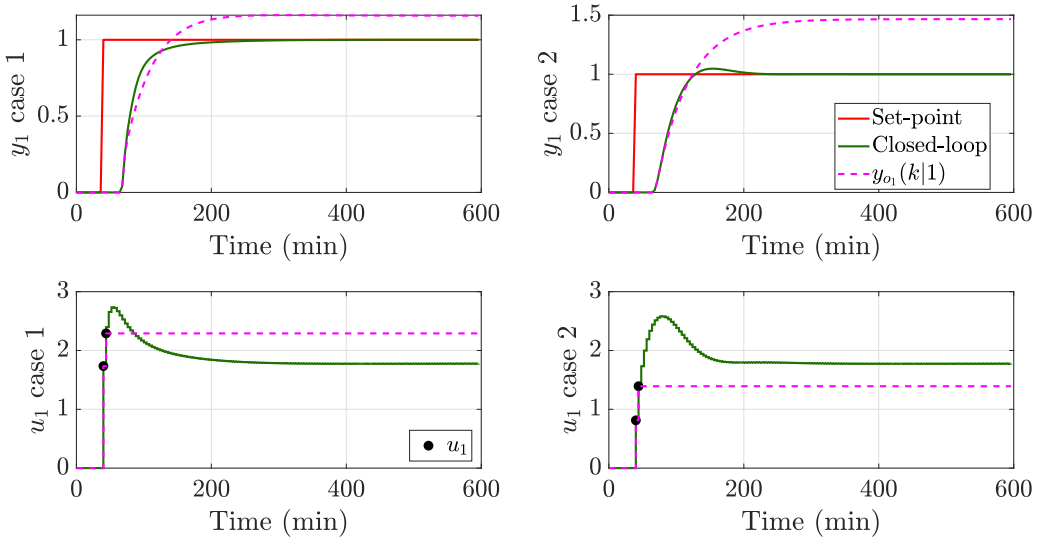


Figure 3.12: Determination of the prediction and control horizons of the HOF output y_1 and input u_1.

The same setpoint has been configured for the GAM algorithm and the plant to consider the internal interaction between the variables of the MIMO system. The lower and upper bounds of the input and the minimum and maximum input increments of the MPC controller are $\mathbf{u_{min}} = [-0.5, -0.5, -0.5]$, $\mathbf{u_{max}} = [0.5, 0.5, 0.5]$, $\mathbf{\Delta u_{min}} = [-0.05, -0.05, -0.05]$, and $\mathbf{\Delta u_{max}} = [0.05, 0.05, 0.05]$. The setpoints are changed to $\mathbf{r} = [0.2, 0.2, 0.2]$ at 70 min, then to $\mathbf{r} = [0.0, 0.4, 0.1]$ at 315 min, then to $\mathbf{r} = [0.1, 0.3, 0.0]$ at 800 min, and finally to $\mathbf{r} = [0.0, 0.0, 0.0]$ at 1600 min, with unknown pulse disturbances of intensity -0.05 on input u_1 from 1100 min to 1120 min, and intensity 0.1 on input u_2 from time 1400 min to 1420 min. Figures 3.15 -

Table 3.1: MPC tuning parameters for HOF study cases

MPCT Scenario	$\mathbf{x}_{dv}$
case 1	$[\underbrace{34}_{p}, \underbrace{2, 2, 3}_{\mathbf{m}}, \underbrace{0.38, 0.08, 0.12}_{\text{diag}(\mathbf{Q})}, \underbrace{0.075, 0.00036, 0.61}_{\text{diag}(\mathbf{W})}]$
case 2	$[\underbrace{8}_{p}, \underbrace{2, 2, 3}_{\mathbf{m}}, \underbrace{0.29, 0.10, 0.08}_{\text{diag}(\mathbf{Q})}, \underbrace{0.27, 0.02, 2.28}_{\text{diag}(\mathbf{W})}]$

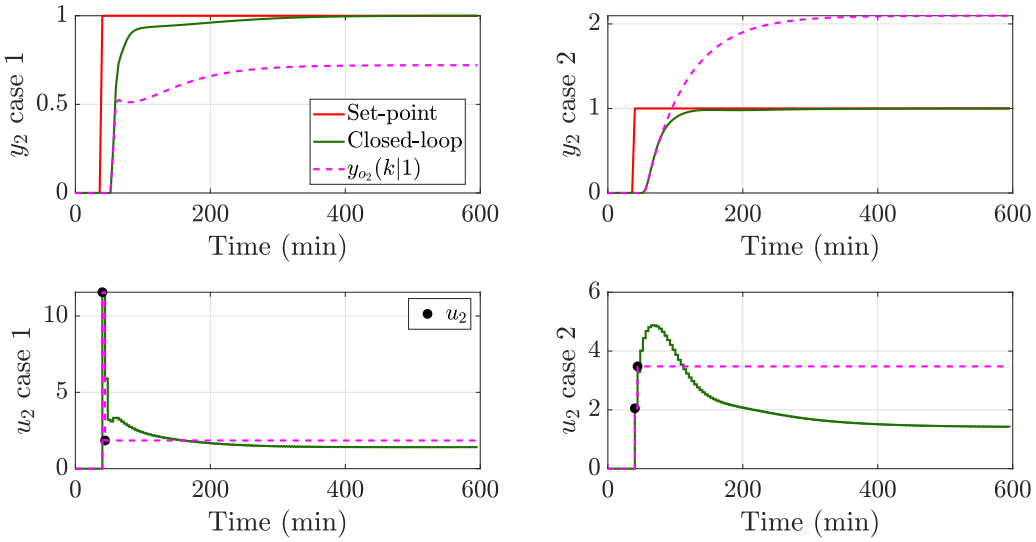


Figure 3.13: Determination of the prediction and control horizons of the HOF output y_2 and input u_2.

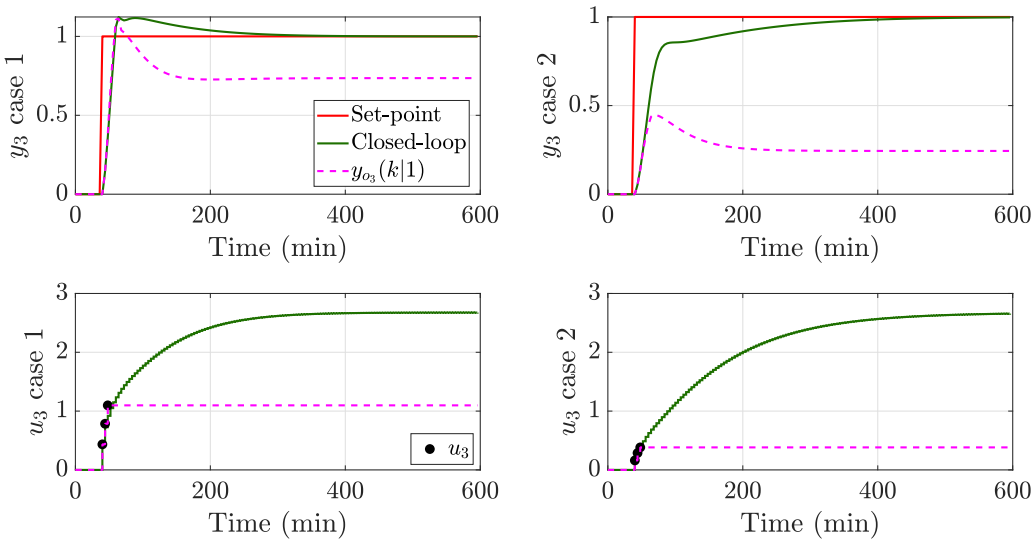


Figure 3.14: Determination of the prediction and control horizons of the HOF output y_3 and input u_3.

3.18 show the behavior of the MPC with tuning parameters shown in Table 4.1 for both cases. Two scenarios are considered for every case: (i) the nominal case with $\epsilon_1 = \epsilon_2 = \epsilon_3 = 0$ and (ii) the modelling error case with $\epsilon_1 = \epsilon_2 = 0.2$, and $\epsilon_3 = 0.3$.

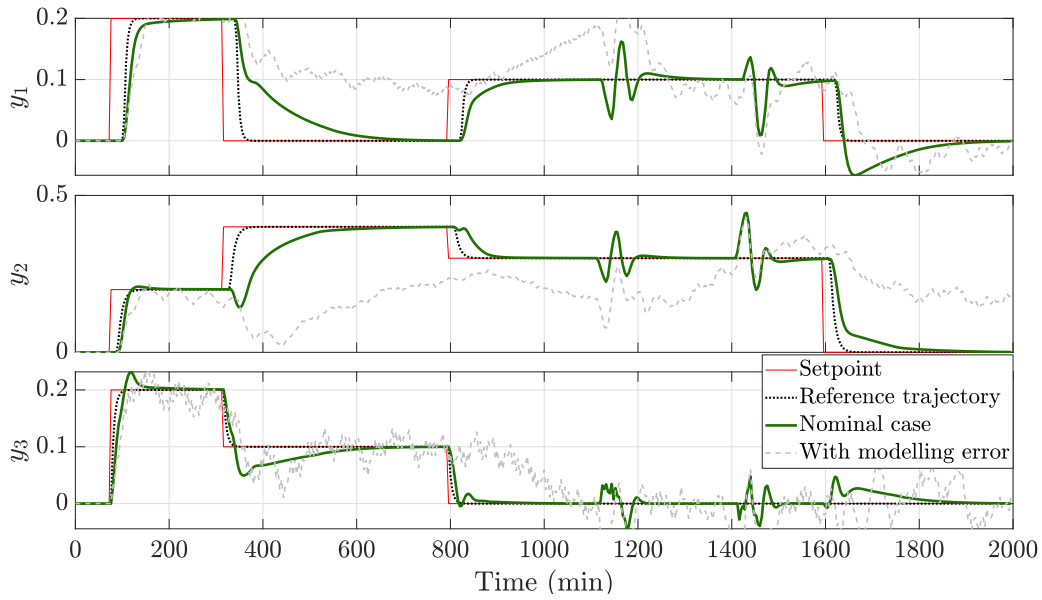


Figure 3.15: Case 1—Response of the HOF outputs to setpoint changes and disturbance rejections.

Both tunings seek to adjust the response with the reference trajectory to attend to the performance established by the user in the nominal case, as exhibited in Figures 3.15 and 3.17. These dynamics are especially evident in the first setpoint change at 70 min, when the response complies with the established accommodation time. However, it is also fulfilled at other simulation points, such as 800 min. In this case study, the setpoint changes were intentionally set in opposite directions, as shown in Figures 3.15 and 3.17 at 315 min, to increase the influence of the interaction between variables. At the points where the interaction is strong, the response cannot adjust to the reference trajectory; however, it manages to track the setpoint without an offset. Moreover, the MPC controller rejects the load of unknown pulse disturbances on the three manipulated variables faster than the tracking accommodation time. Additionally, it is observed that y_3, which has the minimal effective time delay, is the variable that suffers most with the effect of the interactions of the MIMO system caused by y_1 and y_2. Therefore, this variable presents the highest overshoot in the reference changes, mainly if these changes are also applied to the other two variables simultaneously.

It is also possible to observe the robustness of the tuning parameters, which are directly related to the performance desired by the user. The robustness is guaranteed only for case 2 because it establishes a conservative desired dynamics compared to

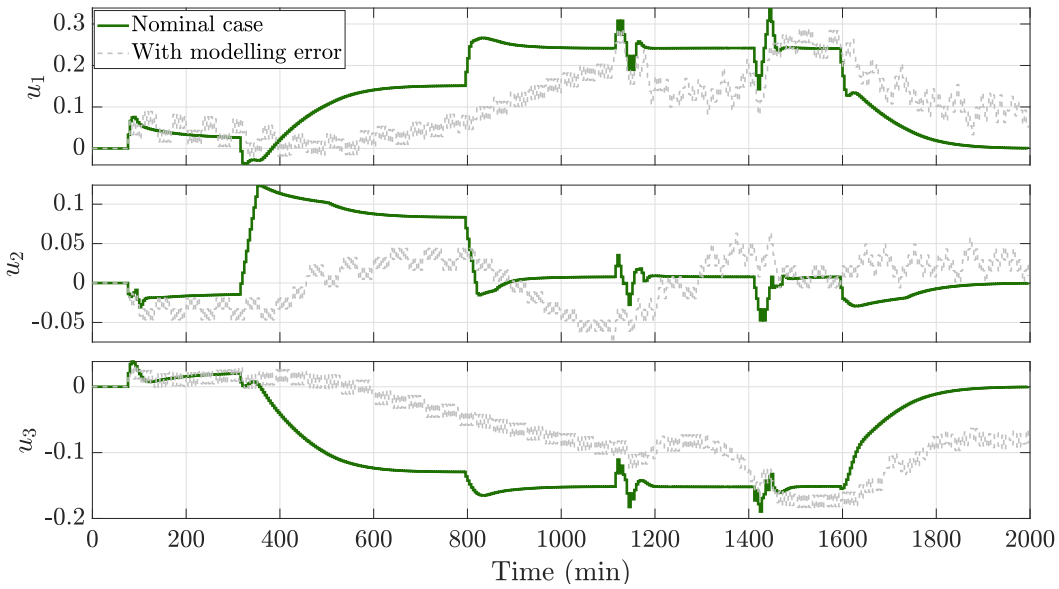


Figure 3.16: Case 1—Manipulated variables of the HOF.

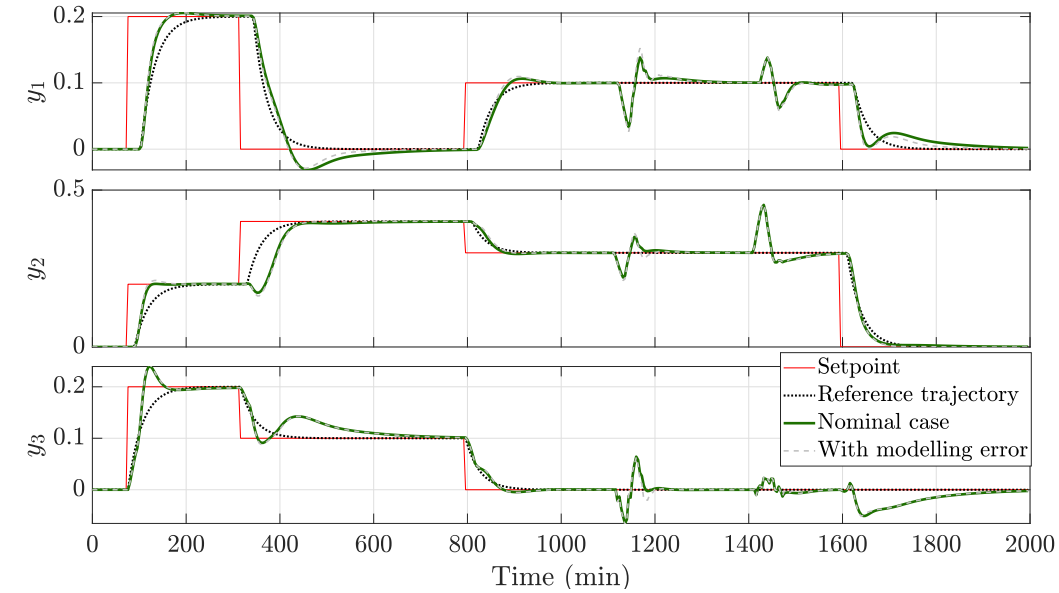


Figure 3.17: Case 2—Response of the HOF outputs to setpoint changes and disturbance rejections.

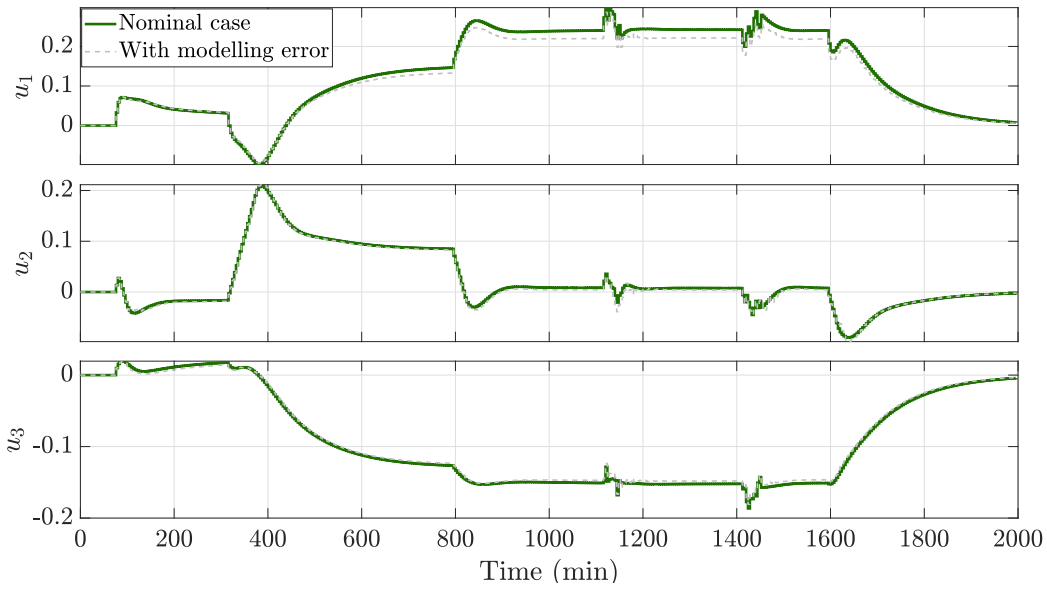


Figure 3.18: Case 2—Manipulated variables of the HOF.

case 1. The MPCT algorithm maintains robustness in case 2 because it searches for the weighting matrices of the objective function using the GAM optimization algorithm for a conservative dynamic that presents a slow change and a moderate control action. This concept can be extended to a practical level, where a family of models can represent the real process so the tuning parameters can be calculated using the worst case.

It is important to remember that robustness is not the exclusive task of the MPCT optimization algorithm. Rather, it can work with the MPC control algorithm to obtain better results in a robust configuration. To demonstrate this, Figures 3.19 - 3.22 show the dynamic behavior of the two cases, further increasing the modeling error for $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1$, using a robust version of the GPC control. This robust version can be achieved using a T filter in the controlled auto-regressive integrated moving average model or a low-pass filter in the optimal predictor stage, known as DTC-GPC (GIRALDO *et al.*, 2021; NORMEY-RICO and CAMACHO, 2007). For a practical understanding of the DTC-GPC algorithm implementation, an illustrative example is provided in Appendix A. For this case, DTC-GPC is implemented with a discrete second-order low-pass filter, as shown in Equation (3.13), using $\alpha = 0.85$ for case 1 and $\alpha = 0.5$ for case 2:

$$F_{r_i}(z) = \frac{(1-\alpha)^2}{(z-\alpha)^2}. \tag{3.13}$$

Implementing a robust control method allows the stabilization of both cases, even with a greater modeling error. The user must establish a compromise between performance and robustness. For case 1, the desired dynamics are very aggressive;

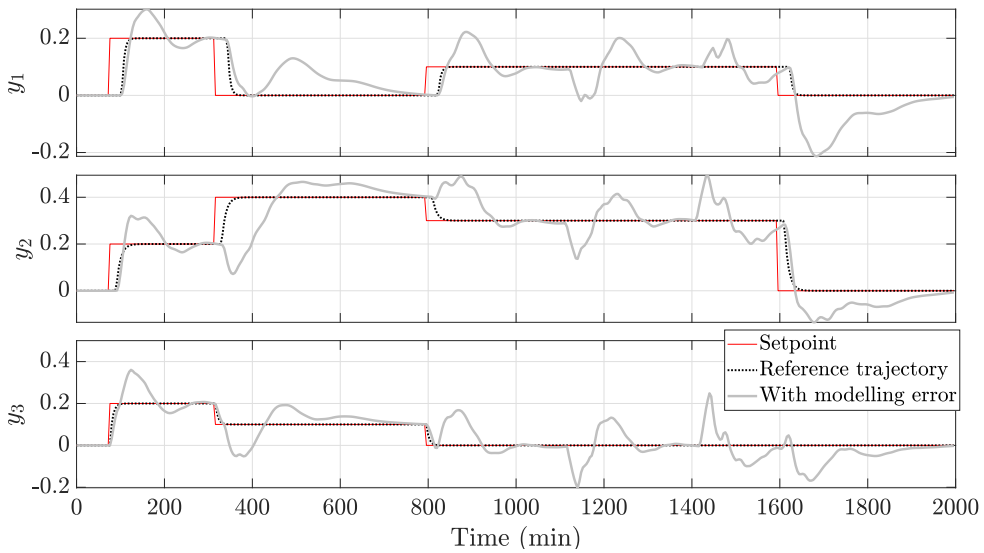


Figure 3.19: Case 1 - Response of the HOF outputs to setpoint changes and disturbances rejections with uncertainties.

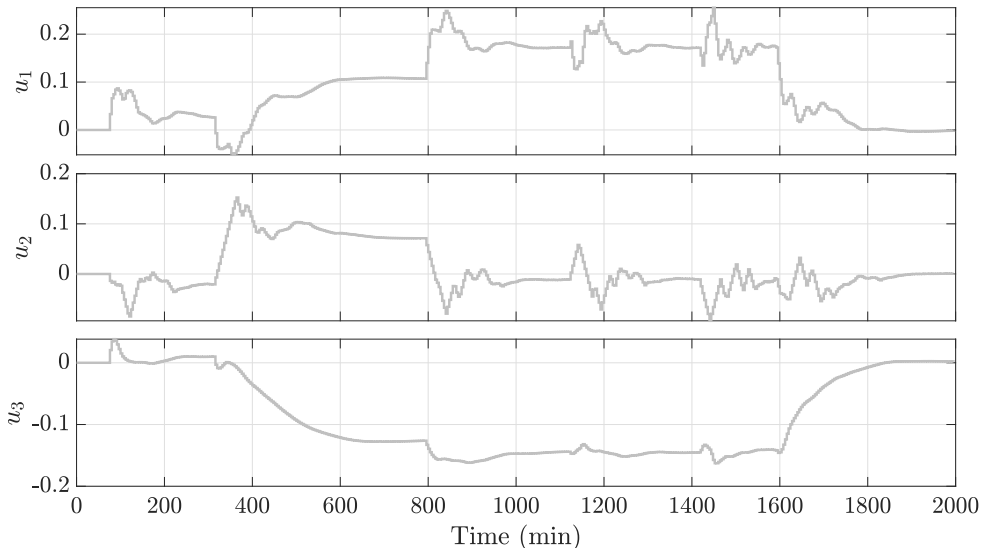


Figure 3.20: Case 1 - Manipulated variables of the HOF with uncertainties.

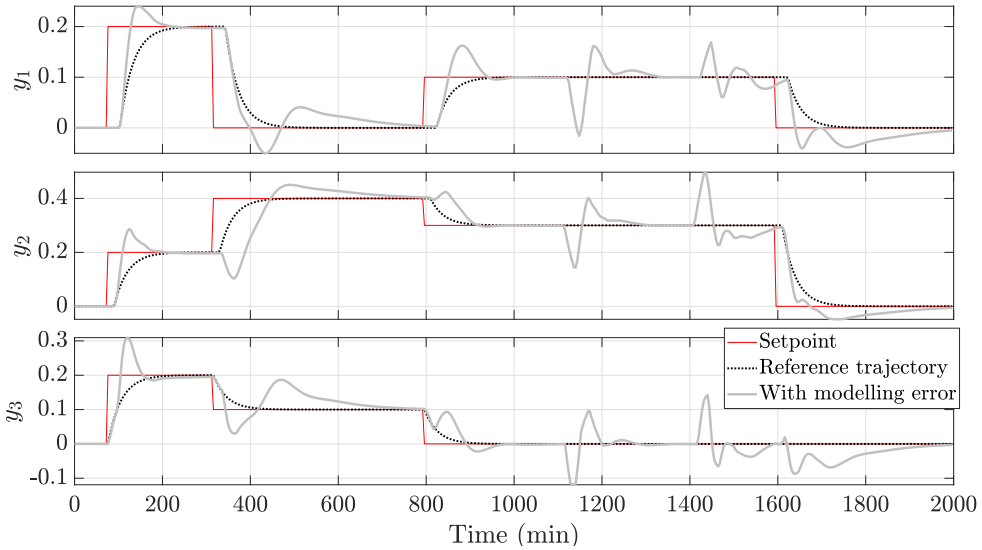


Figure 3.21: Case 2 - Response of the HOF outputs to setpoint changes and disturbances rejections with uncertainties.

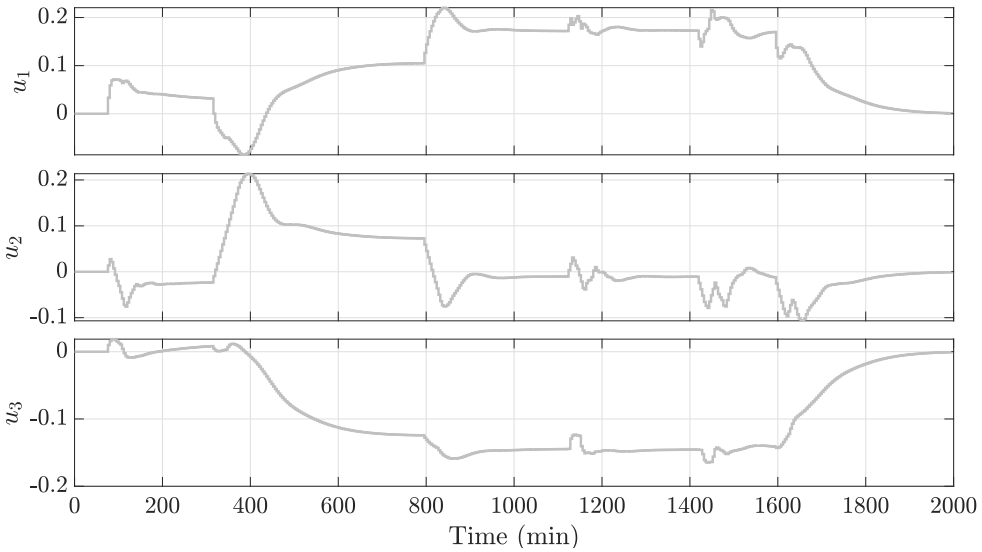


Figure 3.22: Case 2 - Manipulated variables of the HOF with uncertainties.

therefore, it is impossible to meet this specification in the robust control project without sacrificing the system's stability. In case 2, the accommodation time is close to the desired trajectory, excluding the high interaction sections of the MIMO system; however, the high degradation of the model prevents a similar behavior to that shown in the nominal case.

In the cases presented above, the tuning parameters of the MPCT are evaluated under scenarios of model uncertainty, which contribute to the robustness of the proposed method.

Given the focus of this work on the tuning of Model Predictive Controllers, the optimization objective J takes on added significance. It serves not only as the cost function that the MPC aims to minimize but also as a primary indicator of the effectiveness of various tuning schemes. Therefore, to assess the performance of different tuning parameters objectively, the accumulated value of J over the simulation time or a specified time horizon is calculated. This aggregated J value serves as the primary metric for performance comparison among different tuning cases, directly aligning the evaluation criteria with the optimization objective of the MPC.

In Table 3.2, the closed-loop responses are compared against the user-defined desired reference trajectories solely through the lens of the MPC objective function J. For this comparative study, both the nominal and robust control scenarios are considered. A global analysis based on the J values reveals that Case 2 outperforms Case 1. Specifically, in the nominal control scenario, Case 2 exhibits a less negative J value, thereby implying a more balanced optimization of the dual objectives: reference tracking and control effort minimization. Similarly, under robust control conditions, Case 2 manifests superior performance as evidenced by a less negative J value, indicating its designed ability to effectively manage uncertainties and follow the desired trajectory while concurrently rejecting load disturbances.

Table 3.2: Accumulated Objective Function J for the Shell Heavy Oil Fractionator

	Accumulated J
Nominal case 1	-39.1665
Nominal case 2	-0.29555
Robust case 1	-388.0707
Robust case 2	-0.3726

3.3.2 The Van de Vusse Reactor — A Square MIMO System with Nonlinear MPC Formulation

A nonlinear continuously stirred tank reactor conducting the well-known Van de Vusse reactions was chosen to test the behavior of the MPC controller using the tuning proposed in this work. In this case, a Nonlinear Model Predictive Control (NMPC) formulation was selected to control the system. The NMPC formulation employed utilizes a Single Shooting method.

The Van de Vusse scheme ($A \xrightarrow{k_1} B \xrightarrow{k_2} C$ and $2A \xrightarrow{k_3} D$) comprises two reactions of the reactant A, producing the desired product B and to the undesired byproducts C and D (GRAICHEN *et al.*, 2004).

The reaction rates parameters k_i, $i = 1, 2, 3$ depend on the temperature, T, and they are represented by the Arrhenius equation:

$$k_i(T) = k_{i0} exp\left(\frac{-E_i/R}{T(^\circ C) + 273.15}\right), \tag{3.14}$$

where E_i, $i = 1, 2, 3$, are the activation energies of the three reactions and R is the universal gas constant.

The process is described by the non-adiabatic model, represented by the following mass and energy balance equations in the reactor:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - k_1(T)C_A - k_3(T)C_A^2, \tag{3.15}$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + k_1(T)C_A - k_2(T)C_B, \tag{3.16}$$

$$\begin{aligned}\frac{dT}{dt} =& \frac{1}{\rho C_p}\left[k_1(T)C_A(-\Delta H_{RAB}) + k_2(T)C_B(-\Delta H_{RBC}) + \right.\\ & \left. k_3(T)C_A^2(-\Delta H_{RAD})\right] + \frac{F}{V}(T_0 - T) + \frac{K_w A_R}{\rho C_p V}(T_k - T),\end{aligned} \tag{3.17}$$

where the concentration of A in the reactor and in the feed are, respectively, C_A and C_{Af}; C_B is the desired output of the concentration of B; the manipulated inputs are the dilution rate, F/V, and the reactor jacket temperature, T_k; V is the constant reactor volume; T_0 is the feed temperature; ρ is the liquid density; C_p is the heat capacity. $Q = K_w A_R(T - T_k)$ is the heat transferred from the reactor to the jacket, where K_w is the heat transfer coefficient and A_R is the surface area for heat transfer; reaction enthalpies are given by $(-\Delta H_{RAB}), (-\Delta H_{RBC}), (-\Delta H_{RAD})$.

The parameter values of the system were obtained from TRIERWEILER (1997)

and are presented in Table 4.3.

Table 3.3: Van de Vusse parameters (TRIERWEILER, 1997)

Parameters	**Value**	**Unit**
k_{10}	1.287×10^{12}	h^{-1}
k_{20}	1.287×10^{12}	h^{-1}
k_{30}	9.043×10^{9}	L/(mol h)
$-E_1/R$	-9758.3	K
$-E_2/R$	-9758.3	K
$-E_3/R$	-8560.0	K
$(-\Delta H_{RAB})$	-4.20	kJ/mol
$(-\Delta H_{RBC})$	11.00	kJ/mol
$(-\Delta H_{RAD})$	41.85	kJ/mol
ρ	0.9342	kg/L
C_p	3.01	kJ/(kg K)
K_w	4032.0	kJ/(h K m^2)
A_R	0.215	m^2
V	10	L
T_k	128.95	°C
T_0	130.0	°C
C_{Af}	5.10	mol/L

The two system inputs, u_1 and u_2, are the dilution rate F/V and the reactor jacket temperature T_k, respectively. The two controlled outputs, y_1 and y_2, are the concentration in C_B and the reactor temperature T, respectively. The input/output variable pairs of the Van de Vusse reactor were established as u_1:y_1 and u_2:y_2. The NMPC sampling period is set to $T_s = 0.05$ h. The prediction horizon was set as $p = 32$ (5-bits), the control horizon was set as $\mathbf{m} = [7, 7]$ (3-bits each) and the tuning horizon was set as $\phi = 60$.

Prior to implementing the MPCT algorithm on the Van de Vusse reactor, it is crucial to establish the scale factors. These are calculated as the span (maximum - minimum) of the variables, ensuring normalization of the quantities involved. For this process, the scale factors for the manipulated variables are [150, 110], for the output variables [1.2, 110], and for the state variables [6, 1.2, 110]. Ensuring these factors are determined at the onset of the controller design and kept constant throughout, promotes numerical stability.

To show the competitive objectives of the GAM formulation presented in Equation (3.4), Figure 3.23 depicts the compromise optimization results in the Pareto frontier for the Van de Vusse problem.

Two cases for the reference trajectory are also considered here, governed by first-order systems, where the static gains are $\mathbf{k}^R = [1, 1]$ and the two time constants

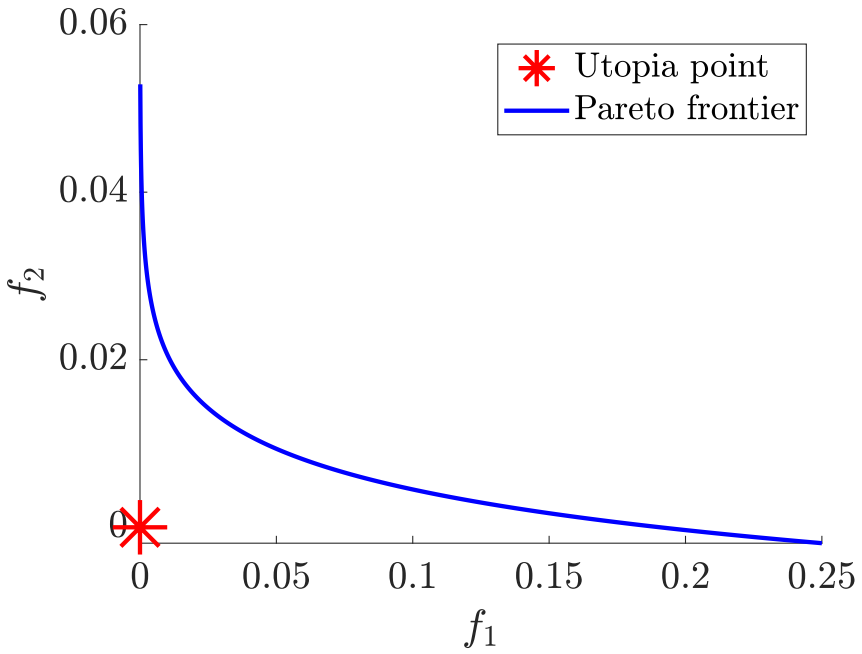


Figure 3.23: Compromise optimization in the Pareto frontier for the Van de Vusse problem.

are ${\boldsymbol{\tau_a}}^R = [0.05, 0.0875]$ hours and ${\boldsymbol{\tau_c}}^R = [0.3, 0.4]$ hours for the aggressive and conservative dynamics, respectively.

One may note that the orders of magnitude between the concentration and the temperature are different; therefore, Equation (A.2) is used to scale the variables between 0 and 1. The relative weight for the GAM was set as $\boldsymbol{\omega} = [1, 1]$. The step references for the VNS algorithm are set as $\mathbf{r}_s = [0.1, 4]$ from the steady-state. Table 3.4 presents the solution of the tuning procedure for both cases.

Table 3.4: Tuning parameters for Van de Vusse reactor

MPCT Scenario	$\mathbf{x_{dv}}$
case 1	$[\underbrace{15}_{p}, \underbrace{2,2}_{\mathbf{m}}, \underbrace{1,1}_{\text{diag}(\mathbf{Q})}, \underbrace{1\times10^{-5}, 1\times10^{-5}}_{\text{diag}(\mathbf{W})}]$
case 2	$[\underbrace{14}_{p}, \underbrace{3,2}_{\mathbf{m}}, \underbrace{0.484, 1.220}_{\text{diag}(\mathbf{Q})}, \underbrace{6.050, 1.44\times10^{-4}}_{\text{diag}(\mathbf{W})}]$

The required computational time was 7 min for case 1 and 140 min for case 2. In Figures 3.24 and 3.25, it is possible to see the approximation of the closed-loop response, $\mathbf{y}(k)$, with the output trajectory at the first optimization, $\mathbf{y}_o(k|1)$, resulting in a suitable horizon size.

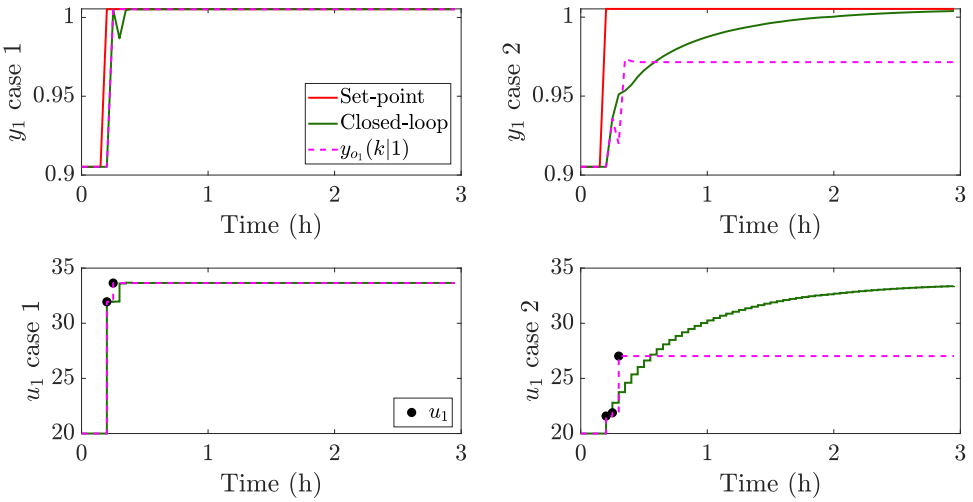


Figure 3.24: Determination of the prediction and control horizons for Van de Vusse Reactor output y_1 and input u_1.

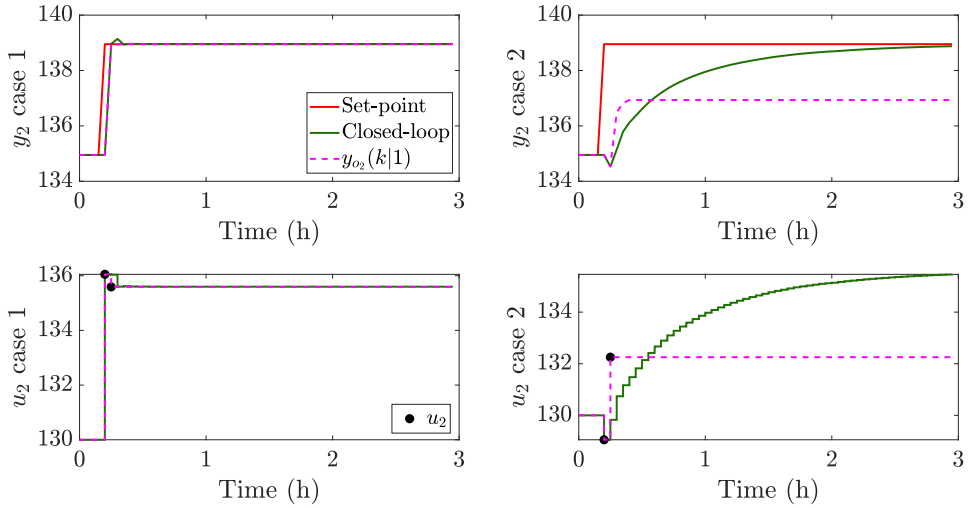


Figure 3.25: Determination of the prediction and control horizons for Van de Vusse Reactor output y_2 and input u_2.

In this case study, the controlled variables are affected by measurement noise and unknown pulse disturbances with an intensity of -5 °C on input u_2 from 6 h to 6.25 h. The response of the tuning of the MPCT on the NMPC can be seen in Figures 3.26 - 3.29. The reference for y_1 changes from 0.9052 mol/L to 1.0 mol/L at 0.3 h and y_2 changes from 134.95 °C to 140 °C at 1.85 h. Additionally, constraints on the process inputs of 0.1 $h^{-1} \leq u_1 \leq 140$ h^{-1} and 90 °C $\leq u_2 \leq$ 200 °C are established.

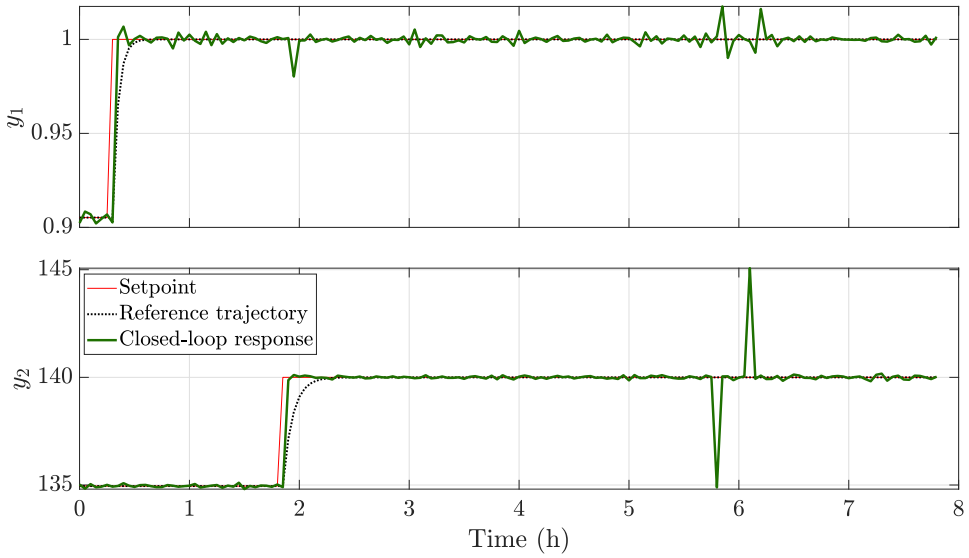


Figure 3.26: Case 1—Response of Van de Vusse reactor outputs to setpoint changes and disturbances rejections.

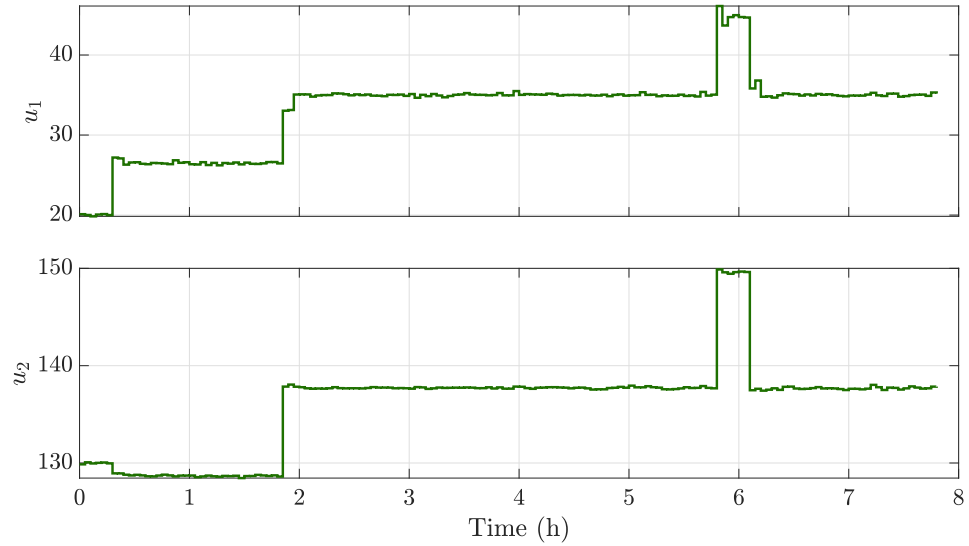


Figure 3.27: Case 1—Manipulated variables of Van de Vusse reactor.

Both tuning strategies exhibit dynamics closely aligned with the user-defined desired reference trajectories. The control mechanisms in both cases successfully

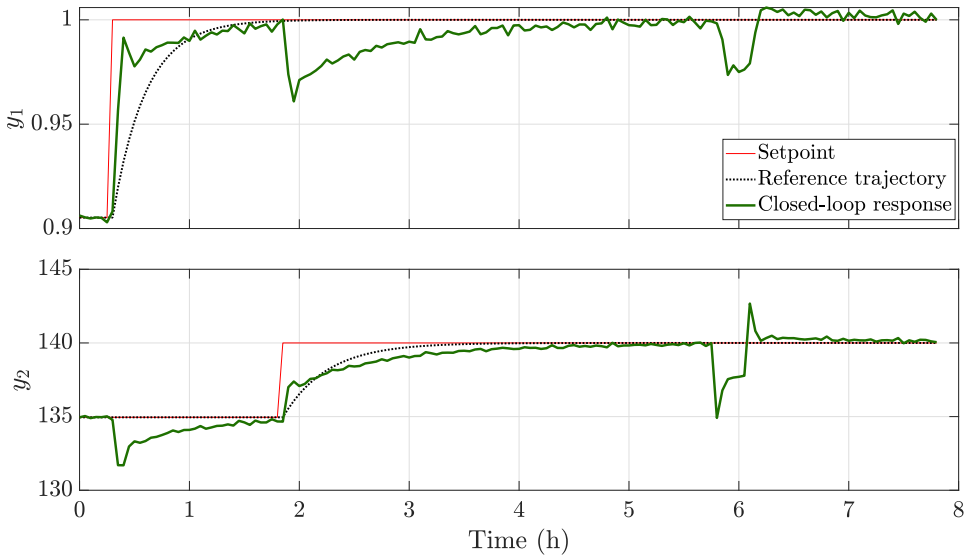


Figure 3.28: Case 2—Response of Van de Vusse reactor outputs to setpoint changes and disturbances rejections.

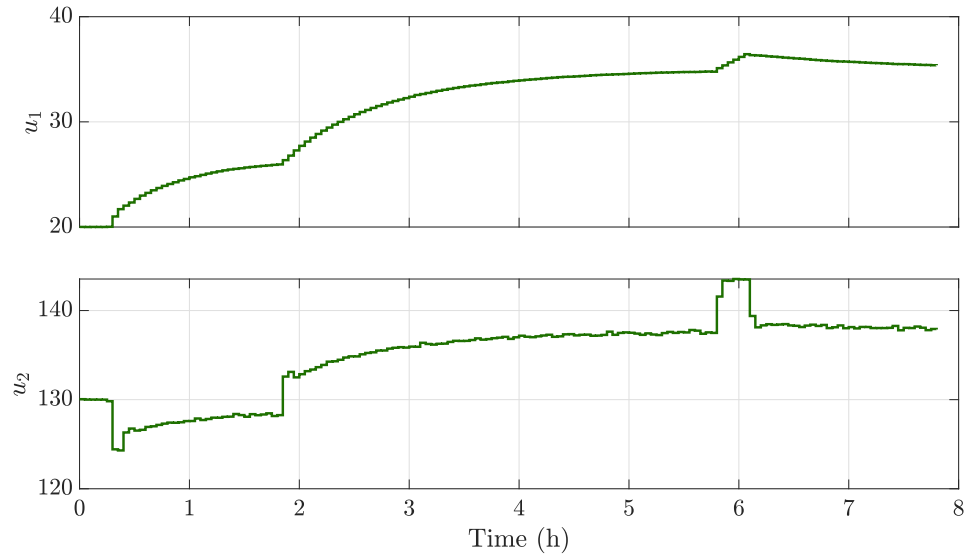


Figure 3.29: Case 2—Manipulated variables of Van de Vusse reactor.

mitigate the noise in the controlled variables and accommodate load disturbances. The value of the multi-objective function J, which combines reference tracking and control effort, is examined to evaluate the different tuning schemes' performance. Table 3.5 shows that Case 1 outperforms Case 2 regarding the accumulated J value. This suggests that Case 1 reaches the setpoint more quickly and offers superior disturbance rejection dynamics, thus fulfilling the dual objectives more efficiently.

Table 3.5: Accumulated Objective Function J for the Van de Vusse Reactor

	Accumulated J
Case 1	2.1538×10^{-5}
Case 2	0.21654

3.3.3 The Shell Heavy Oil Fractionator — A Non-square MIMO System with Linear MPC Formulation

In order to evaluate the MPCT's performance in tuning MPC controls applied to non-square systems, a full configuration of the Shell Heavy Oil Fractionator (FHOF) as depicted in Figure 3.30 is examined. This system constitutes a distillation column presenting significant control challenges due to interaction between inputs and outputs as well as long time delays. The original system involves 7 monitored variables and 5 inputs, of which 3 are manipulated variables, and 2 are disturbances. The objective is to control the top output composition (y_1), the side output composition (y_2), the top temperature (y_3), the top reflux temperature (y_4), the side draw temperature (y_5), the mid reflux temperature (y_6), and the bottom reflux temperature (y_7) by manipulating the top draw (u_1), side draw (u_2), and bottom reflux rate (u_3), while rejecting disturbances provided by the mid reflux rate (d_1) and top reflux rate (d_2).

Each channel of the model is represented by a first-order function with time

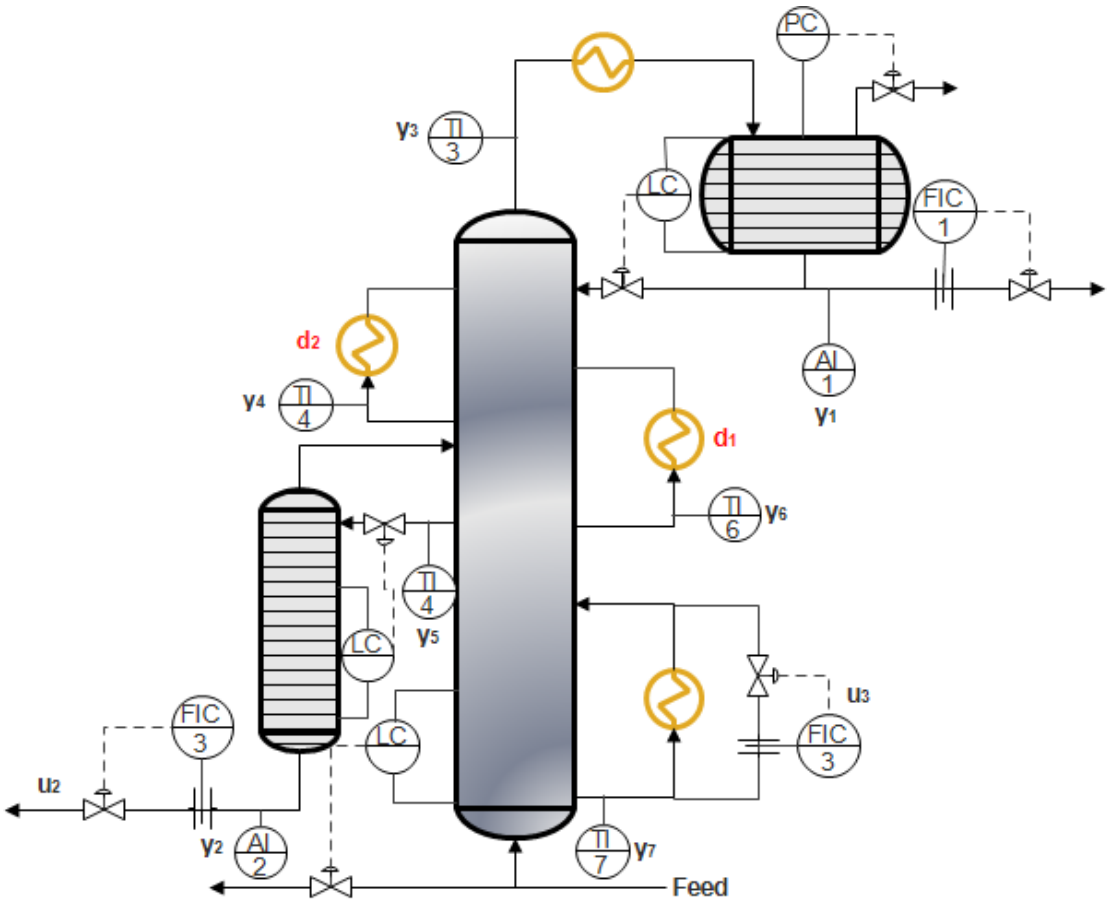


Figure 3.30: Representation of the Shell heavy oil fractionator.

delay, expressed as:

$$\mathbf{P_n}(s) = \begin{bmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} & \frac{1.20e^{-27s}}{45s+1} & \frac{1.44e^{-27s}}{40s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.9e^{-15s}}{40s+1} & \frac{1.52e^{-15s}}{25s+1} & \frac{1.83e^{-15s}}{20s+1} \\ \frac{3.66e^{-2s}}{9s+1} & \frac{1.65e^{-20s}}{30s+1} & \frac{5.53e^{-2s}}{40s+1} & \frac{1.16}{11s+1} & \frac{1.27}{6s+1} \\ \frac{5.92e^{-11s}}{12s+1} & \frac{2.54e^{-12s}}{27s+1} & \frac{8.10e^{-2s}}{20s+1} & \frac{1.73}{5s+1} & \frac{1.79}{19s+1} \\ \frac{4.13e^{-5s}}{8s+1} & \frac{2.38e^{-7s}}{19s+1} & \frac{6.23e^{-2s}}{10s+1} & \frac{1.31}{2s+1} & \frac{1.26}{22s+1} \\ \frac{4.06e^{-8s}}{13s+1} & \frac{4.18e^{-4s}}{33s+1} & \frac{6.53e^{-1s}}{9s+1} & \frac{1.19}{19s+1} & \frac{1.17}{24s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} & \frac{1.14}{24s+1} & \frac{1.26}{32s+1} \end{bmatrix}$$

In this case study, the main novelty lies in the application of the MPCT algorithm to non-square systems, thereby extending its potential uses. This objective is accomplished by incorporating the slack variable, ϵ_k, from the MPC's soft constraints as defined in Equation (2.33), into the GAM's objective function. In contrast to previous studies, where the focus of the MPC controller was to track a setpoint based on a reference trajectory, this study adopts a different strategy for cases with fewer degrees of freedom (specifically, 7 controlled variables for 3 manipulated variables). The control strategy is set to operate within defined ranges, allowing temporary minor violations of constraints using the slack variable but endeavoring to keep the controlled variables within their designated ranges during steady-state operation. In this context, the reference trajectory for the MPCT algorithm is designed to guide the controlled variables to stay within their permissible ranges. This is accomplished by assigning zero weights in the $\mathbf{Q}$ matrix for those variables that operate within these ranges. As a result, the system outputs are allowed to deviate from the reference trajectory, provided they remain within the soft constraints. Normally, soft constraints are applied to the controlled variables, while hard constraints are enforced on the manipulated variables.

The outputs y_1 and y_2 should remain within ± 0.005 specifications, while y_3, y_4, y_5, y_6, and y_7 should remain within ± 0.5 in steady state. All manipulated variables have hard constraints of ± 0.5.

The process model is appropriately scaled using the diagonal matrices $\mathbf{L} = \text{diag}[0.4401; 0.2319; 0.6265; 0.5431; 0.6006; 0.2069; 0.3942]$ and $\mathbf{R} = \text{diag}[0.2640; 0.1351; 0.1156; 0.7819; 0.4665]$ where the first three diagonal values of matrix $\mathbf{R}$ correspond to the manipulated variables, and the last two diagonal values are associated with the measured disturbances.

The scaled gain matrix for the nominal case is given by:

$$\mathbf{K} = \begin{bmatrix} 0.4705 & 0.1052 & 0.2992 & 0.4129 & 0.2956 \\ 0.3300 & 0.1792 & 0.1851 & 0.2756 & 0.1980 \\ 0.6053 & 0.1397 & 0.4007 & 0.5682 & 0.3712 \\ 0.8488 & 0.1864 & 0.5088 & 0.7347 & 0.4536 \\ 0.6548 & 0.1931 & 0.4327 & 0.6152 & 0.3531 \\ 0.2218 & 0.1169 & 0.1563 & 0.1925 & 0.1130 \\ 0.4557 & 0.2354 & 0.3282 & 0.3513 & 0.2317 \end{bmatrix}. \tag{3.18}$$

Setting the scale factors before employing the MPCT algorithm for the FHOF is crucial. These factors normalize the quantities involved and are derived as the span of the variables. In this case, the scale factors for the manipulated variables are set as [1,1,1], for the output variables as [0.01,0.01,1,1,1,1,1], and for the measured disturbance variables as [0.5,0.5].

For the discretization of the process transfer function, a zero-order hold method is employed with a sampling period of $T_s = 4$ minutes. The parameters for the MPCT are established as follows: the maximum prediction horizon p is set to 128, equivalent to 7 bits; the maximum control horizon m is set to 15, corresponding to 4 bits; and the tuning horizon ϕ is fixed at 200. The relative weight for GAM is defined as $\boldsymbol{\omega} = [0.0001, 0.0001, 1, 0.5, 1, 0.5, 1]$.

First-order plus dead-time systems govern the reference trajectory of the MPCT with a static gain denoted by $\mathbf{k}^R = [-1, -1, 1, 1, -1, -1, -1]$. The associated dead-time is $\mathbf{L}^R = [27, 14, 0, 0, 0, 0, 0]$, and the time constant for all outputs are set at $\tau^R = 50$. To generate the reference trajectory, the reference transfer function is stimulated with steps represented by $\mathbf{u}^R = [0.0150, 0.0150, 1.5, 1.5, 1.5, 1.5, 1.5]$, which is thrice the magnitude of the maximum constraint of the plant's output. This specific stimulation was chosen through experimental simulation of the process to establish a reference dynamic that aligns with the magnitude appropriate to each controlled variable.

The tuning parameters for the MPC controller, determined through the use of MPCT, were found to be $p = 23$, $m = 2$, and $\mathbf{W} = [0.012341,\ 9.999\times10^{-6},\ 1\times10^{-5}]$. The required computational time was approximately 12 minutes.

As depicted in Figures 3.31 and 3.32, the close approximation between the closed-loop response, $\mathbf{y}(k)$, and the output trajectory from the initial optimization, $\mathbf{y}_o(k|1)$, can be observed. This showcases the effectiveness of the selected horizon size in capturing the system dynamics.

Figures 3.33 - 3.34 show the MPC by band behavior. In these simulations, it was hypothesized that all inputs and outputs start at zero. However, at time t = 100 min, the system was subjected to a step disturbance of $d_1 = d_2 = 0.5$. As

can be seen in these figures, the controlled variables successfully remain within the

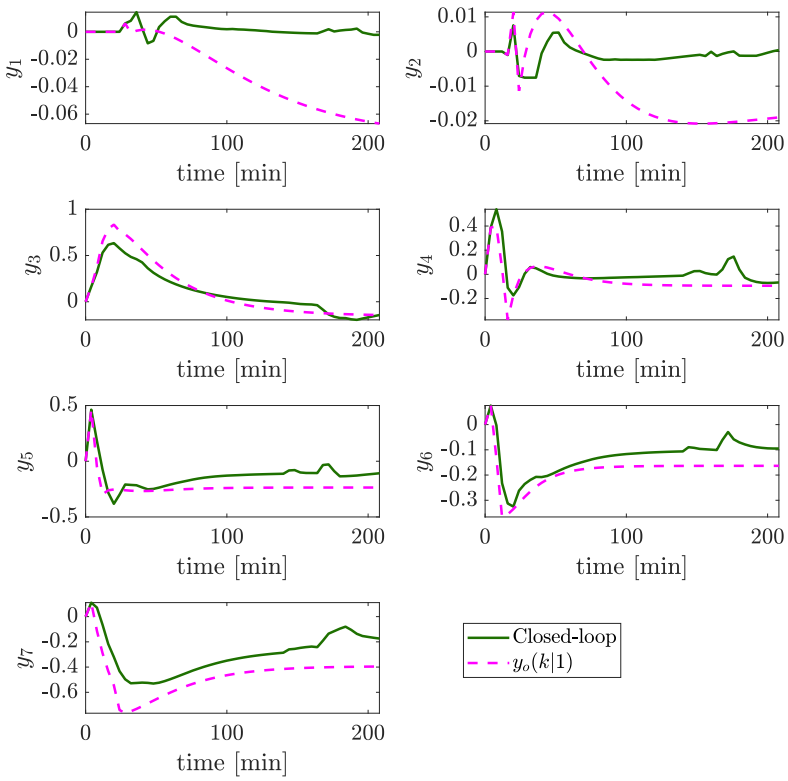


Figure 3.31: Determination of the prediction horizon of the FHOF.

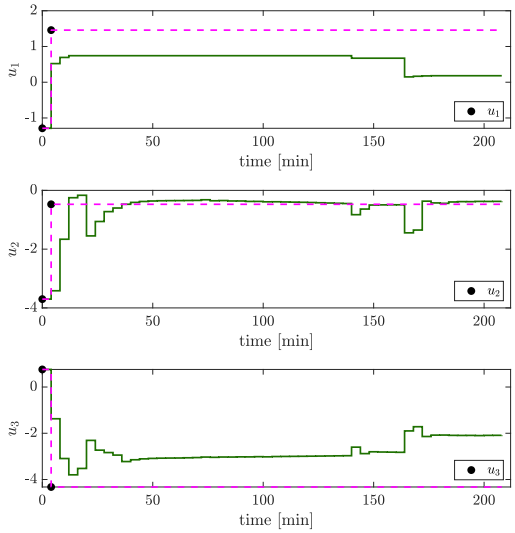


Figure 3.32: Determination of the control horizon of the FHOF.

tracking zone region when the system reaches the steady state, confirming the effectiveness of the proposed methodology in achieving satisfactory process performance.

The controller manages to keep the variables y_1 and y_2 within the bands, which present the most significant restrictions. The reference trajectory was instrumental in determining the controller parameters so that the dynamics could follow this reference. This objective is largely achieved in most of the trajectories. However, the control's priority is to keep all variables within the band, necessitating some variables to stabilize over the soft constraints, as seen in the cases of y_1, y_6, and y_7.

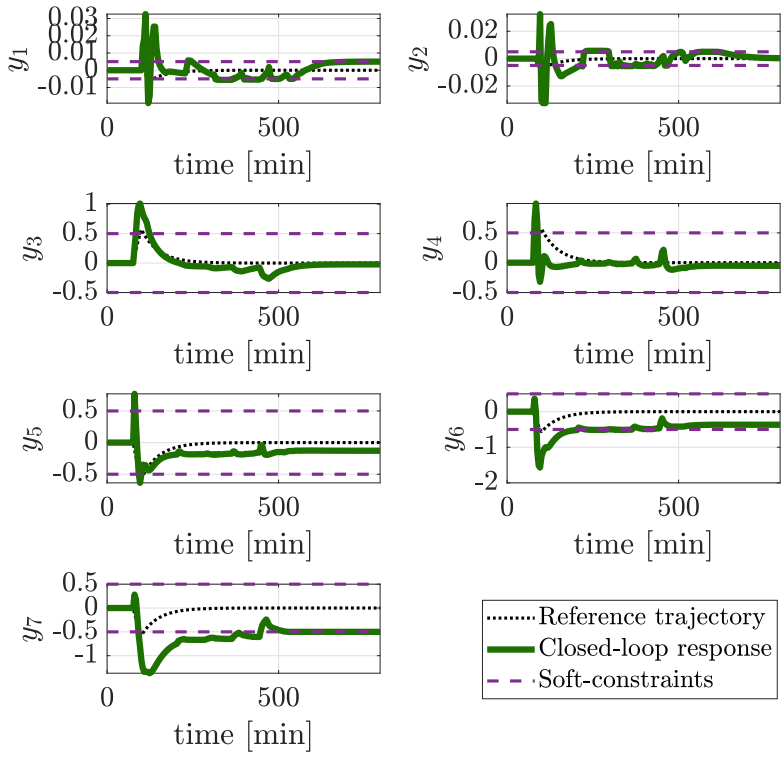


Figure 3.33: Controlled variables of the FHOF.

3.3.4 Production of Butyl Lactate in a Pilot Plant at the National University of Colombia — Linear MPC Formulation

This case study examines the effectiveness of the MPCT applied to a linear MPC implemented in a pilot plant at the National University of Colombia. The plant's purpose is to assess the production of butyl lactate from lactic acid and butanol. The kinetic and thermodynamic information integral to this process can be found

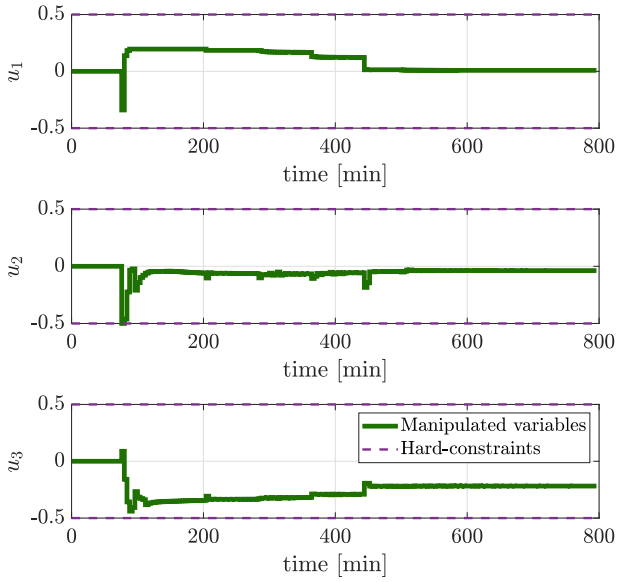


Figure 3.34: Manipulated variables of the FHOF.

in GARCÍA *et al.* (2021) and VELANDIA *et al.* (2021). The plant's representation is a phenomenological model implemented in Simulink and validated with previous experimental runs in the pilot plant and literature data. This model replicates the functionality of a multistage reactive distillation column.

The reactive distillation column (RDC), depicted in Figure 3.35, is theoretically divided into N equilibrium stages for modeling. The first stage is the decanter, the second stage is the condenser, and the N^{th} stage is the reboiler.

Model Assumptions

The modeling of the reactive distillation column has the following assumptions:

- The Non-Random Two-Liquid (NRTL) model is employed for modeling the liquid phase, while the ideal-gas law represents the vapor phase.
- There is no resistance to internal and external diffusive transport over the catalyst, and there is no resistance to transport over the fluid phases. Thus, phase equilibrium is achieved homogeneously.
- A pseudo-homogeneous kinetic model is assumed.
- In the Decanter, the NRTL model is utilized for predicting liquid-phase behaviors, and the k-value method is used to ascertain compositions in the extracted and refined flows.

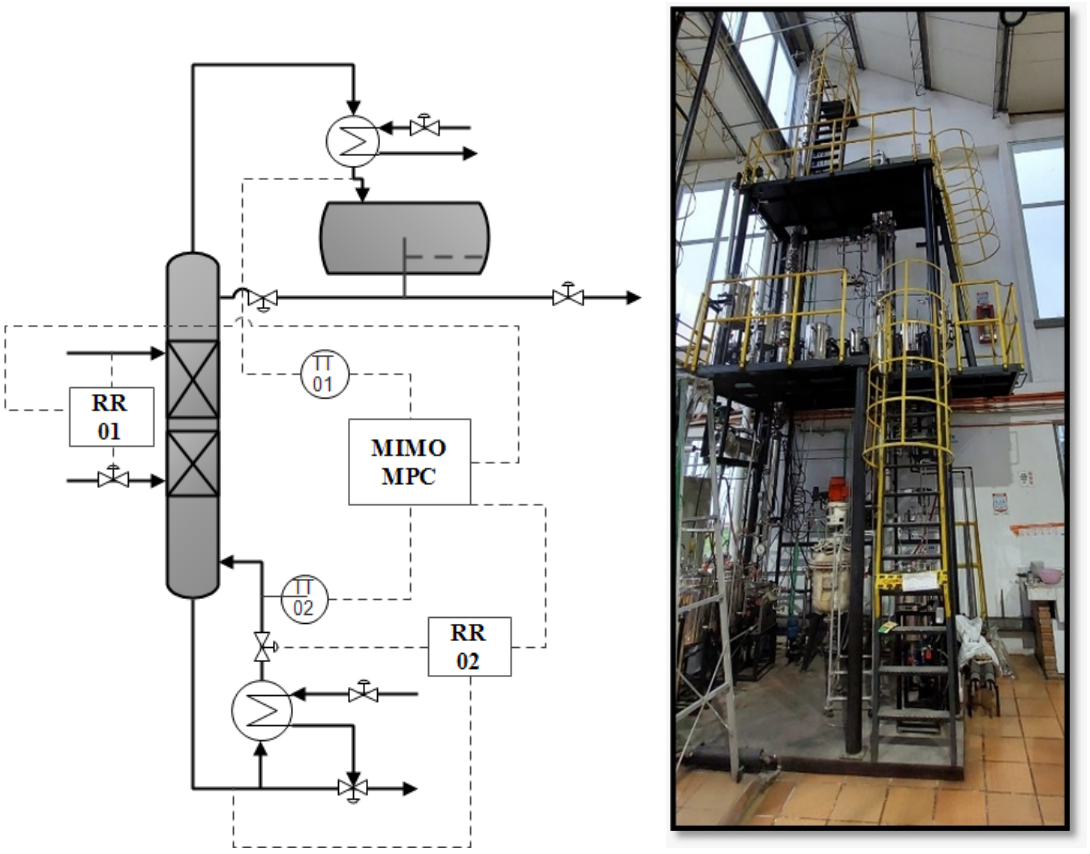


Figure 3.35: Representation of the multistage reactive distillation column.

- The presence of oligomers is neglected.
- The reboiler and condenser are modeled as ideal stages.
- Variations in kinetic and potential energy are ignored.
- Equilibrium is assumed to be controlled by kinetics.
- Physical properties are constant.
- There is no heat loss from the column.
- Column pressure drop is neglected.
- Two phases (vapor-liquid) and a reaction in the liquid phase are considered.

Within the primary focus of this study lies the challenge of accurately tuning a linear MPC controller for a 2×2 system. The manipulated variables, represented as u_1 and u_2, correspond to the feed and boil-up ratios, commonly called "RR". The primary objective is to control these ratios to effectively manage the condenser and reboiler temperatures, represented as y_1 and y_2, respectively. All controller tuning

and subsequent operations are based on the validated model of a pilot plant-scale reactive distillation column.

Managing these variables proves to be a significant challenge due to the system's non-linearity, the over-dimensioning of specific column equipment, and the presence of multiple steady states within this process. Therefore, the condenser temperature, y_1, is maintained within an operational range between 359.3 K and 361 K, and the reboiler temperature, y_2, is set between 430 K and 440 K. These regions are implemented to ensure the production of the target product, Butyl Lactate, with a desirable composition exceeding 93%. Additionally, rigid constraints on the controlled variables are established to ensure the stability of the column: the feed ratio, u_1, varies between 0.1185 and 0.1449, and the boil-up ratio, u_2, ranges from 13.3850 to 26.77.

The simplified model in the form of a transfer function is given by:

$$\mathbf{P_n}(s) = \begin{bmatrix} \dfrac{2.687 \times 10^{-5}s + 8.43 \times 10^{-9}}{s^2 + 0.0001108s + 2.271 \times 10^{-8}} & \dfrac{-1.495 \times 10^{-5}s + 7.807 \times 10^{-9}}{s^2 + 0.0004505s + 7.777 \times 10^{-8}} \\ \dfrac{0.0161s - 7.563 \times 10^{-6}}{s^2 + 0.0001978s + 1.688 \times 10^{-8}} & \dfrac{2.591 \times 10^{-5}}{s + 2.531 \times 10^{-5}} \end{bmatrix}. \tag{3.19}$$

MPC Controller

The process model is appropriately scaled by the minimum conditioning number using the diagonal matrices $\mathbf{L} = \text{diag}[0.7075; 0.0064]$ and $\mathbf{R} = \text{diag}[0.0156; 0.6286]$.

For the nominal case, the scaled gain matrix is defined as:

$$\mathbf{K} = \begin{bmatrix} 0.0041 & 0.0446 \\ -0.0446 & 0.0041 \end{bmatrix}. \tag{3.20}$$

The manipulated variables' scale factors are designated as [0.0132; 13.3850], and the controlled variables as [1.5; 40]. The nominal values for the manipulated variables are defined as [0.1396; 13.3850], while for the controlled variables they are [359.8552; 407.2594]. Regarding the controller constraints, the controlled variable $\mathbf{y}$ is subject to $\pm[1, \infty]$. The manipulated variables are bounded within specific limits: the lower bound $\mathbf{u}_{\min}$ is defined as [0, 0], and the upper bound $\mathbf{u}_{\max}$ is [0.0132, 13.3850]. It is pertinent to note that the scale factors, nominal values, and controller constraints are pre-multiplied by $\mathbf{L}$ and $\mathbf{R}^{-1}$, following the procedure elaborated in Appendix A.

The tuning parameters giving by the MPCT include a sampling period of $T_s = 1000$ s, a prediction horizon of $p = 100$, a control horizon of $\mathbf{m} = [2, 4]$, and the semi-definite weighting matrices are $\mathbf{Q} = \text{diag}([0;\ 0.1127])$ and

$\mathbf{W} = \text{diag}([0.6013;\ 0.4027])$. As the system output can operate within a range, as long as it abides by flexible constraints, the first element of the $\mathbf{Q}$ matrix diagonal is set to zero. Consequently, the flexible constraint violation penalty weight is $\rho_\epsilon = 10000$.

In Figure 3.36, the adjustments made by the MPCT algorithm to fit the prediction and control horizons can observed. Notably, the closed-loop dynamics align closely with the dynamics of the MPC's first optimization, thus providing a reliable estimate of the process trajectory and ensuring appropriate sizing of both horizons.

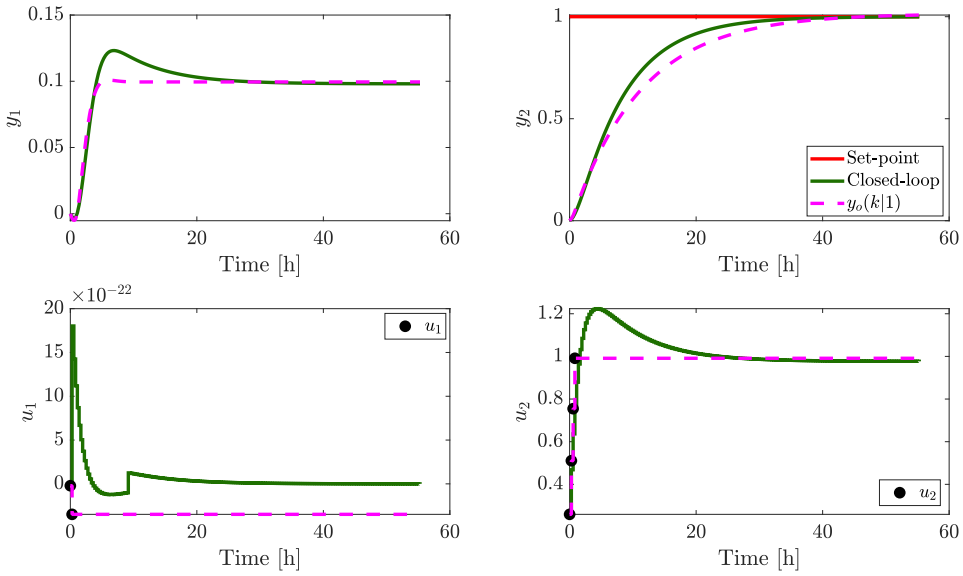


Figure 3.36: Determination of the prediction and control horizons for RDC.

Figure 3.37 illustrates the control response of the condenser and reboiler temperatures by manipulating the feed and boil-up ratios. The results from the linear MPC control implementation on the reactive distillation column show the system initiating from a steady state, with the condenser temperature at 359.86 K and the reboiler temperature at 407.26 K. An initial setpoint is set for the reboiler temperature at 430 K. Based on this specification, the linear MPC controller adjusts the manipulated variables, reducing the feed ratio to approximately 0.12 and increasing the boil-up ratio to 22. This adjustment results in a temperature increase of 0.5 K in the condenser, ensuring the controlled variable remains within the established range. At the same time, the reboiler temperature reaches the desired setpoint around the 15-hour mark.

Following 50 hours, a new setpoint of 440K is set for the reboiler temperature, and the MPC increases the boil-up ratio until it stabilizes at 23, simultaneously reducing the feed ratio during the reboiler temperature transient, but it eventually settles around 0.12. The reboiler temperature reaches the new setpoint around the

65-hour mark. The condenser temperature remains within the soft constraints at all times, a significant consideration given the over-dimensioning of the equipment controlling this variable, making the column control highly challenging.

These results prove that the linear MPC controller efficiently regulates the condenser and reboiler temperatures according to system specifications, even when the system exhibits high nonlinearities and potentially multiple steady states.

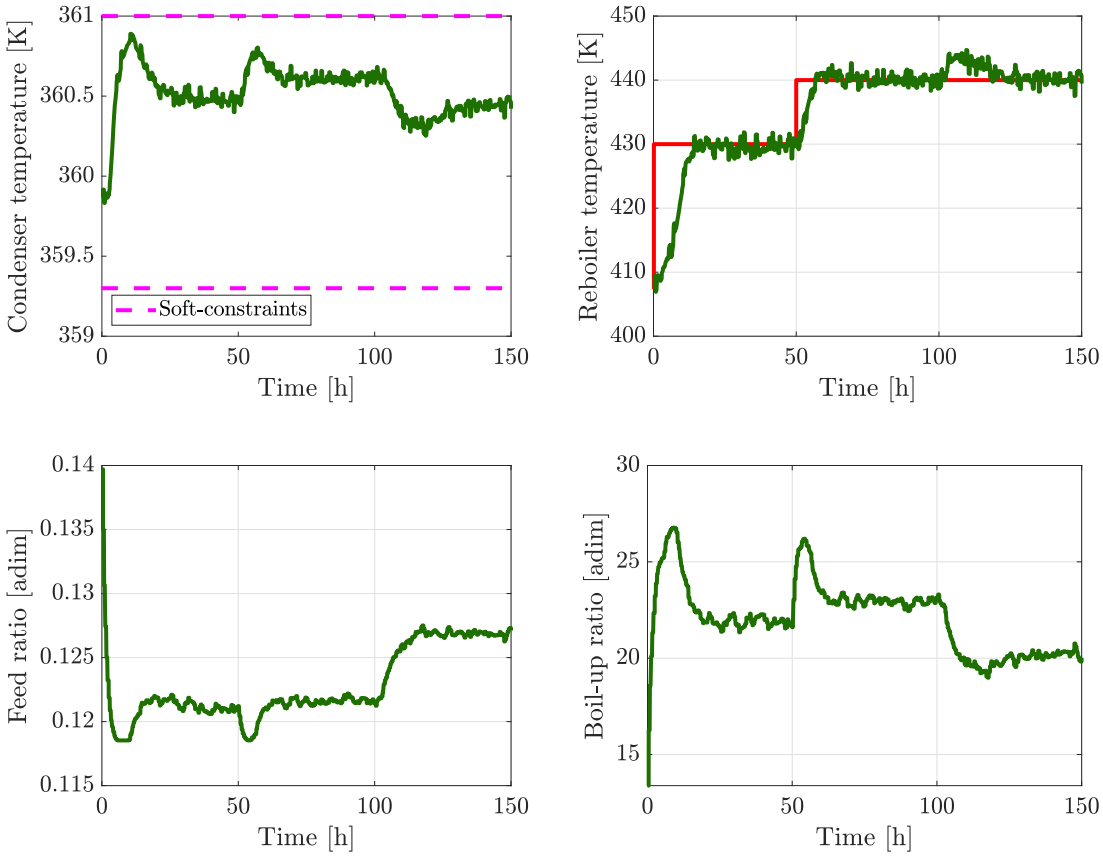


Figure 3.37: Responses of the MPC on the reactive distillation column.

3.4 Final Remarks

This chapter has introduced the Model Predictive Control Tuning Approach (MPCT). This novel hybrid optimization method uses both the Goal Attainment Method (GAM) and the Variable Neighborhood Search (VNS) to adjust the predictive and control horizons and the weight matrices of an MPC.

The strength and versatility of the MPCT algorithm have been showcased through simulation case studies applied to different process systems. These cases included both square and non-square MIMO systems, linear and nonlinear MPC formulations, demonstrating the wide-ranging applicability of the MPCT approach.

Notable cases included the subsystem of the Shell Heavy Oil Fractionator, the Van de Vusse Reactor, and the production of Butyl Lactate in a pilot plant at the National University of Colombia.

These diverse applications illustrated the MPCT's effectiveness in tuning MPCs applied to complex processes, achieving desirable steady states, maintaining system variables within safe operation bands, and responding efficiently to setpoints or disturbance conditions changes. These case studies have substantiated the strength of MPCT in handling systems with inherent non-linearities, over-dimensioned equipment, and potential multiplicity of steady states, which are frequently encountered in real-world process systems.

Chapter 4

Filtered Smith Predictor Monitoring, Diagnosis, and Self-tuning due to Unmeasured Abrupt Load Disturbance or Model Plant Mismatch

The second part of the doctoral thesis focuses on providing real-time diagnostics for the current control loop state in single-input, single-output systems, specifically when using dead-time compensators (DTC) for filtered Smith predictors (FSP). This chapter introduces a novel algorithm that offers not only model error (MPM) and unmeasured disturbance (UD) detection but also auto-tuning of the robustness filter to ensure system stability. This algorithm opens up the potential for utilizing internal model parameters for model updates or the implementation of other control methods documented in the literature, thus facilitating adaptive control.

4.1 Introduction

Monitoring and maintaining the performance of control systems is a critical task in various industrial applications. Over the years, researchers have proposed various techniques for monitoring and diagnosing the performance of control systems. DING and LI (2021) provide a comprehensive review of existing techniques for control performance monitoring and degradation recovery, including some new results and future perspectives. SHEIKHI *et al.* (2021) propose an approach to monitoring the performance of nonlinear control systems by projecting a nonlinear predictive generalized minimum variance control onto a second-order Volterra structure. The

proposed control scheme can handle constrained problems and use more adjustment knobs to improve closed-loop performance. BOTELHO *et al.* (2021) propose an MPC model monitoring and diagnosis approach for non-square systems, which can improve the performance and reliability of such systems. These authors propose a methodology that combines residual analysis, fault diagnosis, and reduced modeling techniques to monitor and improve control system performance. The results show that the proposed methodology can detect performance problems in the MPC model and adjust the model to improve the stability and efficiency of the control system.

A fundamental factor in model-based controllers assessment in the closed-loop control is identifying the source of the model's predictive capacity loss, which could be a model-plant mismatch (MPM) or an unmeasured disturbance (UD). An MPM occurs when the dynamic model cannot correctly represent the evolution of the process variables, and a reidentification is required. A UD happens when any input variable was not considered in the process model (BOTELHO *et al.*, 2016a). Both MPM and UD can negatively affect the system's performance, and distinguishing between the two is not trivial. The MPM may result in performance degradation or instability of the control system, as the model fails to accurately represent the real system dynamics. This mismatch can be caused by system behavior changes over time or inaccuracies in the model's initial identification. On the other hand, the UD may not induce instability in linear systems; however, in nonlinear systems, a UD could drive the plant to an unstable operating point where the controller may not stabilize the system. Still, they can induce sustained errors or significant deviations in the process outputs, adding extra, unmodeled input to the system. A classic example of MPM occurs when some process actuator begins to lose its capacity to manipulate the system due to lack of maintenance or wear of its mechanical parts. This causes the relationship between the inputs and outputs of the process to be not the same as those represented by the identified model. On the other side, the UD can occur, for example, by entering some noise on the signal of some sensor that affects the capacity of action of the controller.

If the process to be controlled has a dominant time delay, the Filtered Smith Predictor (FSP), proposed by NORMEY-RICO and CAMACHO (2007), is an efficacious DTC can be used to control stable, integrating, and unstable processes. The FSP employs a process model to generate a control signal, effectively compensating for dead time in the process. Therefore, the knowledge of the predictive capacity of the FSP given by the internal model and the detection of the UD effects is vital for the controller performance and robustness assessment.

Every predictive structure that uses the open-loop model as a reference, such as the FSP or MPC, to forecast the future dynamic of the process and make corrective feedback actions need to analyze two main different scenarios (BOTELHO *et al.*,

2015), (i) faster tuning: the controller is more sensitive to the MPM on the transient response and less sensitive to the steady state; (ii) slower tuning: the entire system response is relevant, and the controller is less sensitive to the MPM due to the slower control action.

BADWE *et al.* (2010) proposed a performance assessment method to a model-based predictive control (MPC). They incorporated the concept that the model error effects on controller performance depend not only on the MPM, but are also a function of the controller tuning and disturbances. The authors proposed identifying the design plant behavior in a closed-loop, known as the design sensitivity function (S_d), to quantify the impact of the MPM on controller performance. As this method requires two data-based model identifications, an expensive procedure, BOTELHO *et al.* (2015) proposed an approach that relies on the nominal sensitivity function for MPM evaluation in an MPC controller. The authors employed the measured data, the simulated outputs, and the complementary sensitivity function (via identification procedures) to estimate the design plant behavior. With this method, the authors could assess the actual control closed-loop performance.

Despite the significant advances in the cited literature, the development of diagnostic, monitoring, and self-tuning algorithms that can detect model degradation or sudden entry of an abrupt disturbance in the Filtered Smith Predictor structure has not been fully addressed. Also, the precise distinction between model predictive capacity loss due to a model-plant mismatch and an unmeasured disturbance remains a challenge in the field. There is an opportunity to enrich the existing performance monitoring and diagnostic techniques and improve their effectiveness in real-life situations. Therefore, this work aims to contribute to this effort, proposing a new methodology to enhance the robustness and effectiveness of control systems.

As the sensitivity transfer function makes it possible to obtain the nominal closed-loop performance (without MPM or UD) and investigate the real effects of model-plant mismatch and unmeasured disturbances. This work proposes an algorithm that establishes a time window and an operating range as tuning parameters over the nominal closed-loop output and the system's output to verify the model's predicted capacity and the presence of unmeasured disturbances. When a setpoint change is implemented, the algorithm detects MPM by verifying an output error. If MPM is detected, the algorithm checks if the system response is oscillatory due to a significant time delay MPM problem. In that case, the algorithm adjusts the FSP robustness filter to avoid a highly oscillatory response and make the controller more conservative. Finally, the algorithm interprets this behavior as an unmeasured disturbance if the controlled variable is in regular operation and goes out of range for more than k samples without setpoint change.

In addition to monitoring the process signals, this approach employs three op-

timization algorithms to (i) determine the possible actual plant parameters; (ii) find the magnitude of the unmeasured disturbance; (iii) adjust the robustness filter, $F_r(z)$, in case of excessive oscillations caused by a high discrepancy between the system delay and the model delay.

Until now, there has been no open literature on diagnostic, monitoring, and self-tuning algorithms that can detect model degradation or the sudden entry of an abrupt disturbance applying specifically in the Filtered Smith Predictor structure. The necessity to tackle this gap in the literature is of paramount importance and should not be underestimated. Poor model quality or unmeasured disturbances can lead to significant performance degradation in the closed-loop control system, especially when the dead-time parameter is affected. Thus, this work aims to fill this gap, particularly in SISO (single input single output) dominant time delay processes, by proposing an algorithm to diagnose, monitor, and self-tune the Filtered Smith Predictor structure under different scenarios, including model-plant mismatches and unmeasured disturbances. This algorithm aims to improve the robustness and effectiveness of the control system, ensuring reliable and stable operation even in the presence of these issues.

4.2 Methodology

The proposed method in this work is based on creating an envelope around the designed closed-loop output $y_d(k)$, referred to as the benchmark output and introduced in section 2.5.6. The width of the envelope is defined as an algorithm parameter. The method detects model discrepancies and abrupt load disturbances by comparing the dynamics with the benchmark signal. Additionally, to guarantee robustness in the control structure, the algorithm adapts the filter $F_r(z)$ if high-risk scenarios exist, especially in the time delay mismatch scenario. The proposed approach has three objectives: (i) to alert the user if an MPM or a UD has occurred, (ii) to show the possible parameters of the actual transfer function for future structure maintenance or to make an adaptive control strategy, and (iii) to adjust the robustness filter automatically in a critical MPM case.

Figure 4.1 depicts the control structure proposed in this work, where $y_n(k)$ represents the disturbance output and $\mathbf{f}$ is the vector containing the parameters used to tune the primary controller with the internal model (if $sc = ON$) and the robustness filter. In this work, based on the presented background, the proposal is to obtain the designed closed-loop output, $y_d(k)$ (without MPM, $\Delta P(z) = 0$, and without UD, $n(k) = 0$), by employing parallel simulation of the designed loop. This approach avoids offline identification of $S_o(z)$ independently of the control complexity used in the structure, such as an MPC. From this nominal closed-loop response,

the performance monitoring and diagnosis of FSP for SISO systems with dominant time delay concerning MPM and UD is developed.

Remark 1. Note that the nominal closed-loop can be executed on the same machine where the controller is implemented or on a different machine to reduce the computational load.

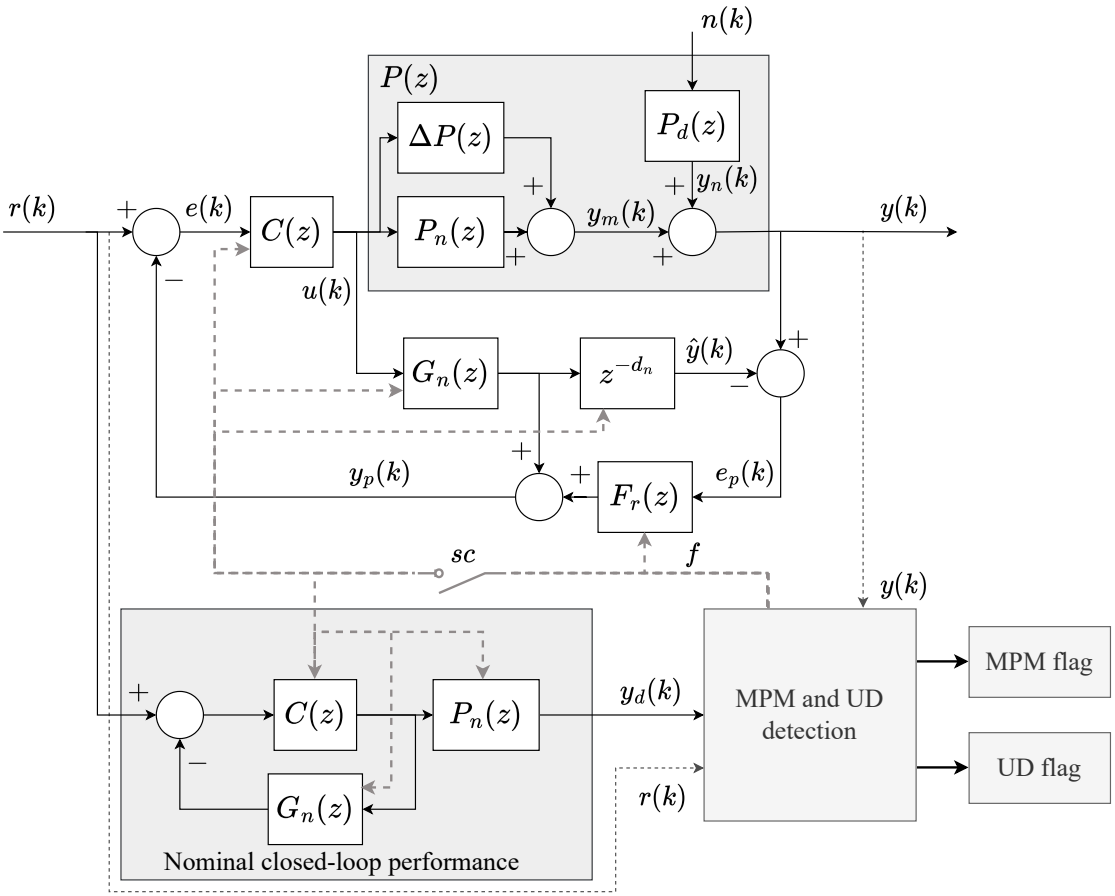


Figure 4.1: Performance monitoring and diagnosis of the FSP structure.

The approach starts with the realistic hypothesis that the controller began its operation with an accurate model so that it would perform well in the first days, weeks, or months. Based on this, in Figure 4.2, it is possible to see the three different scenarios independently: (a) ideal case, $P(z) = P_n(z)$ and $n(k) = 0$; (b) UD case, $P(z) = P_n(z)$ and $n(k) \neq 0$; (c) MPM case, $P(z) \neq P_n(z)$ and $n(k) = 0$. All scenarios have noise on the sensor signal. Notably, these responses represent a generalized system, and the depicted discrepancies illustrate potential effects on the system. Each specific system can exhibit a wide array of dynamics, and as a result, the impact of MPM or UD may differ. Because of the vast diversity of systems and the wide range of possible operating conditions, it is vital to understand that these representations are simplified models that help one conceptualize potential performance issues.

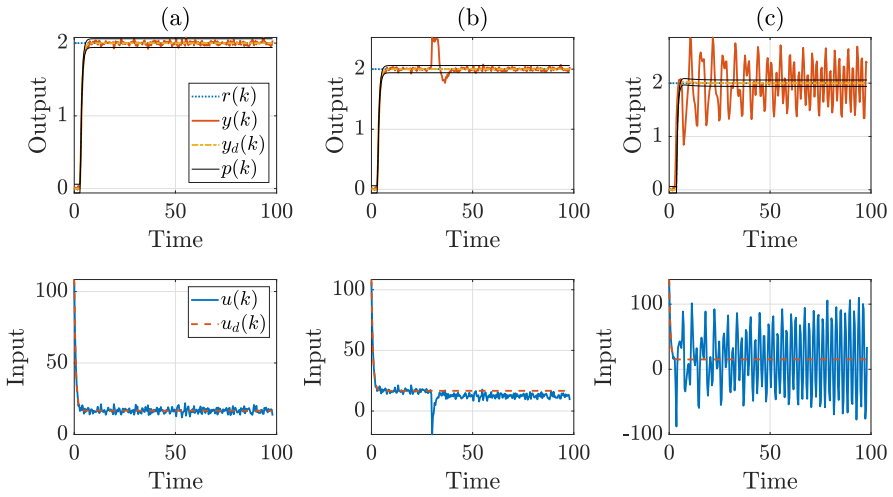


Figure 4.2: Scenarios: (a) ideal case; (b) UD case; (c) MPM Case.

- In ideal case (a), the controller operates with an accurate model and no unexpected disturbances. This scenario serves as a reference for the system's nominal performance, and any deviations from this behavior may indicate the presence of MPM or UD.
- In the UD case (b), the system experiences unexpected disturbances. Although the plant dynamics remain the same as the nominal model, the control action differs due to the unmeasured disturbance. It is possible to use this information to detect and diagnose the presence of UD in the system, as the plant dynamics go outside the envelope without a setpoint change.
- In the MPM case (c), the plant dynamics changed due to process changes or modeling errors. This scenario can be challenging to diagnose because it requires identifying the nature and magnitude of the changes in the plant dynamics. However, the proposed method can still detect and diagnose MPM by comparing the system's dynamics with the benchmark output.

The details of the proposal are described below.

4.2.1 MPM Detection

A previous study of the system's dynamic behavior is the first stage in designing an adequate control system. This knowledge allows tuning the controller parameters to meet the desired specifications. Therefore, the discrepancy between the plant and the model can be considered low at the beginning of the operation. Based on this, a p-band is established over the benchmark signal as a tuning parameter. This p-band must be wide enough to contain the signal of the controlled variable with its noise

at the beginning of the control system operation, that is, when the MPM and UD are low, as shown in scenario (a) in Figure 4.2. A time window, ϕ, is established in the algorithm as a second tuning parameter. During that time window, the analysis of the MPM of the control structure will be carried out. ϕ needs to be large enough to capture the transient and steady-state of the system.

Over time, the model is expected to lose its predictive capacity, causing the controller's performance to decline. Since the setpoint strongly correlates between $y_d(k)$ and $e_0(k)$, this approach can detect the MPM whenever a setpoint change is inserted into the system.

When the setpoint change occurs, the algorithm captures the number of samples configured in the time window ϕ. Once all the samples are collected, the MPM is identified by applying the Algorithm 3.

Algorithm 3: Check if $y(k)$ is within p-band of $y_d(k)$

Input: $y_d(k)$, $y(k)$, p, ϕ
Output: Whether $y(k)$ is within p-band of $y_d(k)$

$e_0(k) \leftarrow y(k) - y_d(k)$;
$upper\_index \leftarrow$ indices of values in $e_0(k)$ greater than p;
$lower\_index \leftarrow$ indices of values in $e_0(k)$ less than $-p$;
$outside\_data \leftarrow$ length of $upper\_index$ plus length of $lower\_index$;
$threshold \leftarrow 0.5\ \phi$;
$std\_dev \leftarrow$ standard deviation of $y(k)$;
$mse \leftarrow$ mean squared error between $y(k)$ and $y_d(k)$;
if $mse > 2\ std\_dev\ OR\ outside\_data > threshold$ **then**
    return *False*;
else
    return *True*;
end

The algorithm compares the two signals $y(k)$ and $y_d(k)$, calculates the difference between them, and identifies the data points that exceed the threshold. The code also calculates the percentage of data points outside the threshold and compares it to a predefined threshold (set to 50%). If the percentage of data points outside the threshold exceeds the predefined threshold, then the code considers the signals significantly different.

In addition, the code calculates the standard deviation of $y(k)$ and the mean squared error (MSE) between the two signals. If the MSE is greater than two times the standard deviation or the percentage of data points outside the p-band is greater than the predefined threshold, the code considers the signals to be significantly different, and an alarm is sent to alert the user about the existence of a significant MPM that degrades the controller's performance.

Using the standard deviation and the MSE is a common approach for detecting

significant signal differences. The standard deviation measures the dispersion of the data points from the mean, and the MSE measures the average squared difference between the two signals. These measures help identify when the difference between two signals is statistically significant.

Additionally, the algorithm can estimate and display the optimized parameters of the transfer function in the internal model by utilizing the following optimization algorithm:

$$J_{MPM}(\mathbf{x}) = \min_{\mathbf{x}} \sum_{k=1}^{\phi} [y(k) - \hat{y}(k)]^2 \tag{4.1}$$
$$\text{subject to}$$
$$\mathbf{L_B} \leq \mathbf{x} \leq \mathbf{U_B},$$

where $\mathbf{x}$ is the decision variable vector containing the parameters of the transfer function model, $P_n(z)$. Note that $P_n(z)$ is a previously known structure. For example, suppose that a first-order plus dead time transfer function, $P_n(s)$, on the continuous domain, represents the model. In that case, the decision variables are $\mathbf{x} = [K, \tau, L]$, where K is the model gain, τ is the time constant, and L is the dead time. The initial condition of the optimization algorithm, $\mathbf{x}_0$, are the same parameters of the transfer function model found at the start of the operation, which is updated if a new model is found. Since this commissioning system constantly monitors the closed-loop control, $\mathbf{x}_0$ provides a good starting point to accelerate the algorithm's convergence. Finally, the search region is bounded by $\mathbf{L_B}$ and $\mathbf{U_B}$, which are the lower and upper limits of the decision variables, respectively. These limits can be established from the initial condition as $x_{0_i}(1 \pm \alpha_i)$ with $i = 1, 2, ..., n_x$, where n_x is the maximum number of decision variables, and α_i is the percentage variation of each parameter of the transfer function model.

Remark 2. $P_n(z)$ is a proper model with any order capable of representing the dynamic process.

The parameters obtained from the process model can be used in various applications, such as implementing any adaptive control technique proposed in the literature. By utilizing the parameters, the controller can adapt itself to the dynamic behavior of the process, resulting in better control performance. Additionally, the parameters can be used for offline configuration of the controller in the plant during future maintenance. This can reduce the plant downtime and improve the control system's efficiency. Overall, the knowledge of the process model parameters can significantly enhance the performance and maintenance of the control system.

4.2.2 UD Detection

One of the primary objectives of process control is to mitigate load disturbances that arise when a system is in a stable state. Load disturbances can include, for instance, changes in the inlet concentration of a reactor with exothermic reactions, changes in road slope while using cruise control in a car, or fluctuations in the water flow rate entering a heat exchanger. Low frequencies are typically dominant in load disturbances and are often modeled using step or ramp signals.

The algorithm introduced in this study can identify sudden load disturbances (as a step function) within a system. The primary objective is to determine if a load disturbance excites the dynamics of the closed-loop system enough to generate an alert for the operator, indicating the potential magnitude of the abrupt load disturbance.

As reported by VERONESI and VISIOLI (2008), to describe different dynamics of the load disturbance, it is supposed that the signal $y_n(k)$ is the step response of a first-order filter $P_d(z)$, namely,

$$P_d(z) = \frac{y_n(k)}{n(k)} = \frac{1-a}{z-a},\ T_s \tag{4.2}$$

where $a = e^{-T_s/\tau_n}$, T_s is the sampling time and τ_n is the filter time constant. Note that the more the filter time constant τ_n is small, the more abrupt the load disturbance over the system is.

After the MPM detection stage has been executed at least once, the diagnostic system is primed for UD detection. Suppose the algorithm identifies that the controlled variable $y(k)$ has exceeded the p-band while superimposed on the signal $y_d(k)$. In that case, this scenario may be interpreted as an unmeasured disturbance (UD) entering the system. To prevent false positives, the algorithm verifies the following conditions:

Condition 1: The signal of the controlled variable $y(k)$ is often contaminated by measurement noise. Therefore, the algorithm employs a filtered signal, $\bar{y}(k)$, rather than the raw signal $y(k)$. Several filtering techniques can be used, and in this study, a moving average filter was selected. A tuning parameter, h_s, determines the filter's horizon and yields the filtered output signal $\bar{y}(k)$:

$$\bar{y}(k) = \frac{1}{h_s}\sum_{m=0}^{h_s-1} y(k-m), \tag{4.3}$$

when the signal $\bar{y}(k)$ deviates from the p-band, and a setpoint change does not exist, the algorithm collects data for the controlled variable $y(k)$ and records the times when $\bar{y}(k)$ remains outside the p-band.

Note: It is imperative to exercise caution when selecting the value of h_s, the filter horizon. An excessively large value for h_s may dampen the system dynamics, leading to attenuated response and potentially the inability to detect disturbances in a timely manner. It is initially suggested to consider a value of h_s that is proportional to the system's time constant (τ) and sampling time (T_s). For this purpose, a factor χ, ranging between 0.5 and 2, can be utilized to adapt h_s as $h_s = \chi\frac{\tau}{T_s}$.

Condition 2: When the signal $\bar{y}(k)$ returns within the p-band, the algorithm waits a while to verify if this signal effectively remains within the p-band. In this way, it is guaranteed that the rejection of the disturbance has been carried out. At that moment, the data collection started in *Condition 1*, is finished, and the abrupt load disturbance is estimated.

Figure 4.3 illustrates an example of an abrupt unmeasured load disturbance. Between the time steps 140 and 150, both the controlled variable $y(k)$ and its filtered mean $\bar{y}(k)$ reside within the p-band. The span between time steps 150 and 159, highlighted in green, signifies a phase where $\bar{y}(k)$ fluctuates outside the p-band, prompting the algorithm to initiate data collection. After this, at time step 159, $\bar{y}(k)$ returns inside the p-band. The algorithm then performs a conditional check based on a combination of multiple criteria. These criteria involve evaluating the recent h_s samples of $y(k)$, as well as computing a mean squared error (MSE) over a window of $\phi/2$ recent samples. If these criteria indicate that the signal has returned to and remained within the p-band, the algorithm concludes that the disturbance has been successfully mitigated. Data collection is then terminated, the magnitude of the disturbance is estimated, and an alarm is issued to inform the operator of the presence of the unmeasured disturbance.

The estimation of the abrupt load unmeasured disturbance is done with the following equation:

$$\begin{aligned} J_{UD}(\mathbf{w}) = & \min_{\mathbf{w}} \sum_{k=1}^{\eta} \left[y(k) - y_d(k) - \hat{y}_n(k)\right]^2 \\ & \text{subject to} \\ & \mathbf{L_B} \leq \mathbf{w} \leq \mathbf{U_B}, \end{aligned} \tag{4.4}$$

where $\mathbf{w} = [\tau_n,\ n(k)]$ is the decision variable vector. A time window, η, is established to capture the time during which the filtered signal $\bar{y}(k)$ oscillates outside of the p-band. This window begins at the moment $\bar{y}(k)$ surpasses the boundaries of the p-band and ends when $\bar{y}(k)$ returns and stabilizes within the p-band. The value of η can vary from system to system and is determined dynamically based on signal behavior. The search region is bounded by $\mathbf{L_B}$ and $\mathbf{U_B}$, which are the lower and upper limits of the decision variables, respectively, where $\tau_n > 0$. The estimation of

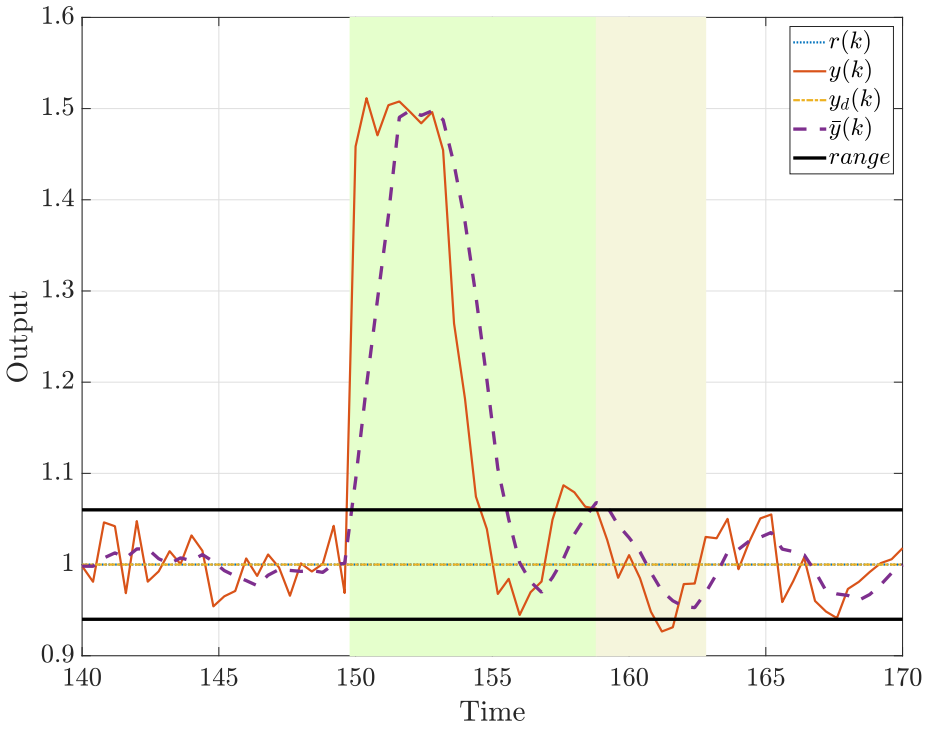


Figure 4.3: Abrupt load unmeasured disturbance.

$y_n(k)$, represented as $\hat{y}_n(k)$, which is illustrated in Figure 4.1 is computed using the following equation:

$$\hat{y}_n(k) = P_d(z)\left[1 - \frac{C(z)P_n(z)F_r(z)}{1 + C(z)G_n(z)}\right] n(k). \tag{4.5}$$

4.2.3 Robustness Self-tuning Filter

When there is a substantial discrepancy between the process and the model, the dynamic response of the closed-loop control can become oscillatory. This is especially true if the process has a dominant time delay. Therefore, the algorithm performs a self-tuning on the robustness filter to smooth the system's response in case the oscillations caused by the MPM appear.

Two conditions must be met when designing the prediction filter, $F_r(z)$: (i) the static gain of the filter should be unitary, i.e., $F_r(1) = 1$, and (ii) the filter should be designed to eliminate the undesired dynamics of the plant model (NORMEY-RICO and CAMACHO, 2009).

The nominal model process, $P_n(z)$, can be written explicitly in terms of the numerator and the denominator, as in Equation (4.6), where the denominator is divided into two parcels, $D_n^+(z)$ and $D_n^-(z)$. The roots $D_n^+(z)$ are the undesired poles of the model plant, and the roots $D_n^-(z)$ are the fast and stable poles of

the model plant. The undesired poles are all those not allowed to appear in the disturbance response.

$$P_n(z) = G_n(z)z^{-d_n} = \frac{N_n(z)}{D_n(z)}z^{-d_n} = \frac{N_n(z)}{D_n^+(z)D_n^-(z)}z^{-d_n} \tag{4.6}$$

The predictor filter can also be written explicitly in terms of the numerator and denominator, such as $F_r(z) = \frac{N_f(z)}{D_f(z)}$. The denominator, $D_f(z)$, may be arbitrarily chosen by the designer with roots within the unitary module circle, considering the desired dynamics in the disturbance rejection and the system's robustness. Note that the roots of $D_f(z)$ are also roots of the disturbance transfer function presented in Equation (2.56). Therefore, slow roots on the filter become the system more robust since the filter module $F_r(e^{j\omega T})$ decay in lower frequencies.

According to NORMEY-RICO and CAMACHO (2009), it can be established that a reasonable choice for the filter's denominator is given by:

$$D_f(z) = (z - \beta)^v, \tag{4.7}$$

where $\beta \in (0, 1)$ and v is an integer such that $v \geq 1$. The parameter v must be selected at least equal to the number of undesired roots of the plant model to be eliminated from $S(z)$ filter presented in Equation (2.57). If v is greater than this number, the robustness increases in the presence of MPM and decreases the disturbance rejection speed.

From Equation (2.57), where $S(z) = G_n(z)\left[1 - F_r(z)z^{-d_n}\right]$, one can isolate the term $1 - F_r(z)z^{-d_n}$ and rewrite it in terms of the filter's numerator and denominator as follows:

$$1 - F_r(z)z^{-d_n} = 1 - \frac{N_f(z)}{D_f(z)}z^{-d_n} = \frac{D_f(z) - N_f(z)z^{-d_n}}{D_f(z)}. \tag{4.8}$$

Note that the numerator of Equation (4.8) is a polynomial. Its roots are meant to cancel out the undesirable poles of the model $G_n(z)$, in other words, the roots of $D_n^+(z)$. Therefore, the polynomial of the filter's numerator can be written as:

$$\frac{D_f(z) - N_f(z)z^{-d_n}}{D_f(z)} = \frac{(z-1)(z-z_1)...(z-z_n)p_r(z)z^{-d_n}}{D_f(z)}, \tag{4.9}$$

here, $z_1, z_2, ..., z_n$ are the undesired poles of $G_n(z)$, $p_r(z)$ is the polynomial residue, and the root in $(z - 1)$ satisfies the criterion $F_r(1) = 1$ (applying the limit of z tending to 1 in Equation (4.9)). The numerator from Equation (4.9) is equated to the poles to be eliminated from the fast model, $G_n(z)$. Thus, the filter's numerator

can be obtained from the following equation:

$$D_f(z) - N_f(z)z^{-d_n} = (z-1)D_n^+(z)p_r(z). \tag{4.10}$$

This procedure guarantees the stability of $S(z)$ since it effectively eliminates the undesired poles from the plant model. This can be seen in the following equation:

$$S(z) = \frac{N_n(z)}{D_n^-(z)} \frac{p_r(z)(z-1)}{D_f(z)}, \tag{4.11}$$

by substituting Equation (4.9) into Equation (2.57), the above equation is obtained, clearly showing how the procedure leads to a stable $S(z)$ by removing the undesired dynamics from the system.

In general, the tuning of the predictor filter of the FSP structure is fixedly defined with an *a priori* design that considers the relationship between robustness and disturbance rejection speed. NORMEY-RICO and CAMACHO (2007); NORMEY-RICO *et al.* (2014) proposed tuning methods for the filter that emphasize robustness or disturbance rejection dynamics, depending on the desired characteristics of the process.

Automatic tuning methods for the robust filter were proposed by DE LIMA and SANTOS (2015) and GIRALDO *et al.* (2016). These methods used online identification of model uncertainties via the fast Fourier transform to adjust the filter parameters and accelerate the disturbance rejection dynamics while ensuring robust stability. In the present work, it is proposed to modify this automatic tuning procedure for the robust filter $F_r(z)$ and integrate it into the diagnosis algorithm of the FSP. Our contribution lies in incorporating a criterion for selecting the parameter that determines the degree of robustness of the filter $F_r(z)$, which enhances the diagnosis algorithm's accuracy and robustness. The estimation of model uncertainties, presented in Equation (2.60), is obtained using the result of Equation (4.1), where $P_i(z)$ is the estimation given by the optimization algorithm, and $P_n(z)$ is the actual internal model of the structure.

Drawing on the concepts introduced in section 3.2.4, the appropriate tuning of the filter $F_r(z)$ to achieve a balance between the robustness of the structure and the speed of the disturbance rejection response is achieved through the definition of the poles in $D_f(z)$. In analytical terms, this involves satisfying Equation (2.62) condition. However, in the context of this proposal, the robustness filter needs to be adjusted conservatively to reduce the oscillations caused by the MPM and allow the dynamic response of the process to stay within the p-band. Consequently, the

robust stability condition is modified to fulfill the following equation:

$$\overline{\delta P_i}(e^{j\omega T_s}) + \gamma \leq dP(e^{j\omega T_s}), \quad 0 < \omega < \frac{\pi}{T_s}, \tag{4.12}$$

where γ is a parameter that affects the system's robust stability margin and can vary between 0 and 1. The closer γ is to one, the greater the increase in the system's robust stability. This work proposes Equation (4.13) as the margin in magnitude measurement to maintain a conservative response:

$$\gamma = \left(1 - \overline{\delta P_i}(\omega_{min})\right) \frac{\omega_{\underline{\text{dP}}}}{10^{\lceil \log_{10}(\omega_{\underline{\text{dP}}}) \rceil}}, \quad 0 \leq \gamma \leq 1 \tag{4.13}$$

where $\omega_{\underline{\text{dP}}}$ is the frequency at which magnitude of the robustness index, $dP(e^{j\omega T_s})$, is minimum, and $\lceil \log_{10}(\omega_{\underline{\text{dP}}}) \rceil$ denotes the ceiling of the logarithm base 10 of $\omega_{\underline{\text{dP}}}$. $\overline{\delta P_i}(\omega_{min})$ is used as a reference to represent the low frequency of the error shape. This means that the robust stability condition is adjusted based on the deviation of the low frequency of the error shape at the frequency that is a certain percentage of the frequency where the magnitude of the robustness index is minimum, concerning the frequency decade that contains this minimum.

The zeros and poles of $F_r(z)$ are estimated using the bisection numerical method to meet the specifications presented in Equation (4.9). For a stable open-loop system, incrementing the filter poles always increases the system's robustness, presenting a monotonic behavior. This behavior is helpful when the bisection method is employed because the technique can find the optimal global value that guarantees a compromise between performance and robustness.

To achieve a disturbance rejection response different from the closed-loop system dynamics, one of the roots of the robustness filter numerator must be set equal to the dominant pole of the closed-loop system. This ensures that the closed-loop filter pole dominates the disturbance rejection speed. For example, let $N_f(z) = (z - \lambda)p_n(z)$, where λ is the dominant pole of the closed-loop transfer function and $p_n(z)$ is the polynomial residue of the filter numerator.

To determine the β parameter, Equation (4.7), of the robustness filter, the bisection algorithm proceeds as follows: since systems that employ the FSP are typically delay-dominant systems, an interval such as $[a, b]$ is defined. This interval is chosen to have two extremes concerning the parameter β. The low extreme value for β is selected in a way that does not guarantee the robust stability of Equation (4.12), and the high extreme value for β is chosen as too robust. The optimal value of the β parameter is reached by reducing the interval in successive halves, verifying that the right endpoint maintains the robust stability condition. This procedure is repeated until reaching a value equal to or less than the stop condition, ε, evaluated by the

difference between a and b. Finally, the right endpoint is selected as the pole of the robustness filter, which presents a compromise between robustness and speed to reject disturbances. Algorithm 4 shows the pseudo-code of the bisection algorithm to find the robust filter.

Algorithm 4: Pseudo-code of the bisection algorithm.

Input : λ, ε, $\overline{\delta P_i}(\omega)$, $C(\omega)$, $G_n(\omega)$, $F_r(\omega)$, ω
Output: $F_r(z)$
// value that does not respect Equation (4.12)
$a \leftarrow 0.001$;
$F_{r_a}(z) \leftarrow get\_filter(\beta = a, \lambda)$; // aggressive filter
$F_a \leftarrow robust\_stability(\overline{\delta P_i}(\omega), C(\omega), G_n(\omega), F_r(\omega), \omega)$; // check Equation (4.12)
// value that respects Equation (4.12)
$b \leftarrow 0.99$;
$F_{r_b}(z) \leftarrow get\_filter(\beta = b, \lambda)$; // conservative filter
$F_b \leftarrow robust\_stability(\overline{\delta P_i}(\omega), C(\omega), G_n(\omega), F_r(\omega), \omega)$; // check Equation (4.12)
if $F_a == 0$ *and* $F_b == 1$ **then**
  while $b - a \geq \varepsilon$ **do**
    $c \leftarrow (a + b)/2$;
    $F_{r_c}(z) \leftarrow get\_filter(\beta = c, \lambda)$;
    $F_c \leftarrow robust\_stability(\overline{\delta P_i}(\omega), C(\omega), G_n(\omega), F_r(\omega), \omega)$; // check Equation (4.12)
    if $F_c == 1$ **then**
      $b \leftarrow c$;
    else
      $a \leftarrow c$;
    end
  end
  $\beta \leftarrow b$;
else
  $\beta \leftarrow b$;
end
$F_r(z) \leftarrow get\_filter(\beta, \lambda)$

The complete repository for the MPM_UD Algorithm can be found on GitHub at the following link: https://github.com/sergioacg/MPM_UD_Algorithm.git.

This repository contains various examples that will be discussed in the following sections. All code has been developed using MATLAB and Simulink, providing a visual representation of MPM and UD alarms directly in the Simulink diagrams. In order to construct the algorithm, S-Functions have been employed. These functions are powerful tools that allow users to create their own blocks in Simulink using MATLAB code. The open-source repository is available to anyone interested in

understanding and implementing the MPM_UD Algorithm in their control system projects.

4.2.4 Algorithm Priorities

Since the FSP commissioning algorithm is divided into three stages, some rules are considered to diagnose the control structure:

1. The diagnostic algorithm only starts operating when the system's first set-point change occurs. In this case, the algorithm prioritizes the execution of the Algorithm 3 and Equation (4.1) to verify the absence of a significant discrepancy between the process time delay and the model time delay. This verification is necessary because a high MPM on the time delay implies an oscillatory response of the system. Consequently, detecting an abrupt UD would be challenging.

2. The UD detection stage is only activated if the MPM detection stage has been executed at least once. The algorithm saves the instant of time in which the first execution of the MPM detection stage was performed. After that, it waits for at least four-time constants before activating the UD detection stage. This allows the process to return to the operating point within the p-band and avoid generating a false positive detection of UD. For example, the process response would be highly oscillatory if the system had a time-delay model discrepancy. These oscillations cause the measured variable to constantly leave the p-band, thus causing false positives of UD detection. However, if the detection stage MPM is activated at least once, this discrepancy can be detected, and the robustness filter $F_r(z)$ is tuned to reduce the oscillations.

4.2.5 Overview of the Proposed Algorithm: MPM and UD Detection, and Auto-Tuning of Robustness Filter

Figure 4.4 provides a comprehensive visualization of the algorithm proposed in this study. This flowchart presents the logical steps the algorithm follows to execute the critical tasks of monitoring, diagnosis, and self-tuning in the context of the FSP control structure, as illustrated in Figure 4.1. The algorithm is designed to emphasize the detection of the MPM, UD, and the self-tuning of the robustness filter.

The monitoring phase is continuous and involves constant observation of the control structure's signals, specifically the reference signal $r(k)$, the benchmark signal $y_d(k)$, and the plant output signal $y(k)$, obtained from the sensor. This constant

monitoring ensures that the system operates within the expected regions during the controller's design phase, thereby mitigating any abnormal behavior.

The diagnosis phase is triggered if the system begins to deviate from the benchmark signal $y_d(k)$ and the signals extend beyond the p-band. Utilizing the monitoring data, a diagnosis is made to identify if the deviation is due to the MPM or an abrupt UD. If either of these scenarios is detected, the algorithm sends an alert to the operator, providing valuable information for scheduling maintenance or possibly tuning the control more conservatively. Furthermore, it offers an estimation of the potential new parameters of the model, which could be beneficial if adaptive control strategies need to be implemented.

Upon the detection of MPM, the auto-tuning phase of the robustness filter is activated. The robustness filter parameters are tuned according to the methodology proposed in this paper, detailed in Algorithm 4. This auto-tuning process, a novel aspect of our proposed algorithm, ensures the internal stability of the control loop when there is a significant MPM, particularly when there is a significant discrepancy in the dead-time parameter.

In summary, the crux of the algorithm is to continuously monitor the system, diagnose potential issues and tune the robustness filter in response to detected anomalies. This proposed method, capable of handling stable systems, integrating or unstable in open-loop, is novel and unmatched in the current literature dealing with the FSP control structure, especially for time-delay dominant processes. The algorithm offers a significant contribution to enhancing the stability and performance of the FSP control structure.

4.3 Case Studies

4.3.1 Application of Proposed Method

A SISO FSP structure was configured in Matlab to illustrate the application of the proposed approach. Table 4.1 shows the real plants and their corresponding models within the FSP structure. Each model presents a mismatch concerning the plant in different dynamic regions. This work presents the systems and their models in continuous time (Laplace domain) to evidence these differences.

The continuous stirred tank reactor with Van de Vusse reactions is chosen for the nonlinear case study. The isothermal Van de Vusse reaction system involves a series and parallel reactions between the feed product A and the obtained product B. The differential equations that govern the concentration inside the reactor are:

$$\frac{dC_a}{dt} = -k_1 C_a - k_3 C_a^2 + (C_{a_i} - C_a)\frac{F}{V} \tag{4.14}$$

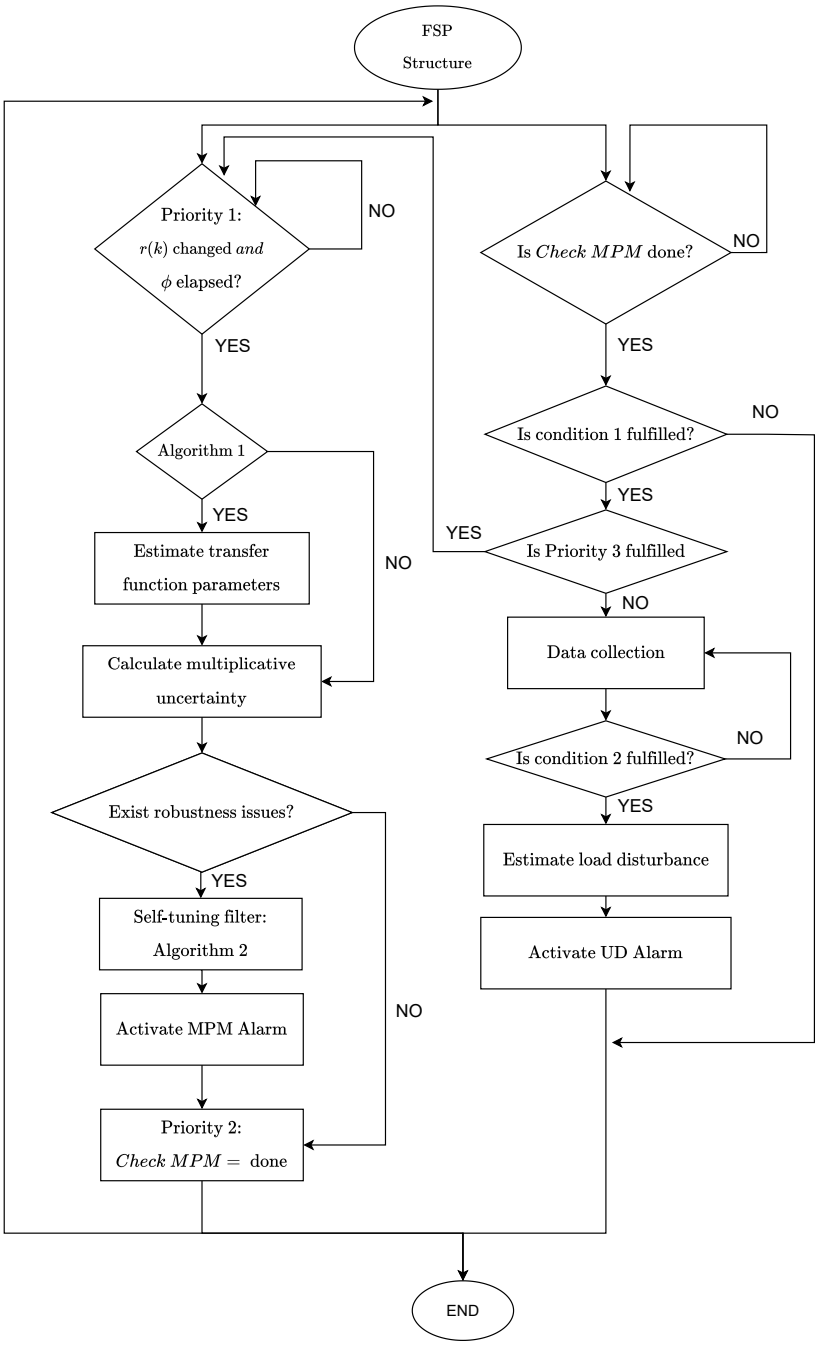


Figure 4.4: Flowchart illustrating the procedure of the proposed algorithm for MPM and UD detection, along with the self-tuning of the robustness filter for the FSP control structure.

Table 4.1: Case Studies

System	Plant	Model
First order	$\frac{0.02\ e^{-3s}}{s+0.1667}$	$\frac{0.0183\ e^{-3.6s}}{s+0.138}$
Non-minimum phase	$\frac{1-s}{s^2+s+1}e^{-3s}$	$\frac{1.2-s}{s^2+1.1s+1.042}e^{-3.6s}$
Oscillatory	$\frac{0.75\ e^{-3s}}{s^2+0.6s+0.25}$	$\frac{0.827\ e^{-2.5s}}{s^2+0.63s+0.2756}$
High order	$\frac{0.75\ e^{-10s}}{(s+0.5)^5}$	$\frac{5.2\ e^{-13s}}{s+0.22}$
Nonlinear	Eqs. (4.14)-(4.15)	$\frac{3.147-1.12s}{s^2+4.643s+5.382}e^{-3.6s}$

$$\frac{dC_b}{dt} = k_1 C_a - k_2 C_b - C_b \frac{F}{V}, \tag{4.15}$$

where C_b is the concentration of B [mol/l], which is the controlled variable, C_a is the concentration of A [mol/l], and C_{a_i} is the concentration of A [mol/l] in the feed. The manipulated variable is the dilution rate F/V [min^{-1}], where F [l/min] is the feed flow rate, and V is the constant reactor volume [l]. The reaction rate constants are given by $k_1 = 5/6$ [min^{-1}], $k_2 = 5/3$ [min^{-1}], $k_3 = 1/6$ [L/(mol min)] (NORMEY-RICO and CAMACHO, 2007).

In the given case study, the system is initially set to an optimal steady state where the production of C_b is maximized. The operating conditions for this optimal steady state are a dilution rate of $F/V = 4/7$ [min^{-1}], $C_a = 3$ [mol/l], $C_{a_i} = 10$ [mol/l], and $C_b = 1.117$ [mol/l]. In this process, the time the analyzer takes to give the measured concentration value is the time delay of the structure.

It should be noted that the controlled variable is the concentration of B, C_b, and the manipulated variable is the dilution rate F/V. These variables are not normalized or represented as deviations from the steady state. Consequently, the gain of the control structure, K_c carries the units of [min^{-1}/(mol/l)]. This non-normalization is a choice made in this study to reflect the actual behavior and control challenges of the system.

The UD estimation was implemented in each case study using the transfer function from Equation (4.2). Notably, for the nonlinear Van de Vusse system, the disturbance was introduced as a step change in $n(k)$, added directly to the process

output while keeping C_{a_i} constant.

The primary controller for the structure is the PI control, where K_c represents the proportional gain, and τ_i represents the integral time. In each case study, the parameters of the PI controllers were adjusted using the MATLAB auto-tuning toolbox. This widely-accepted toolbox employs well-established methods for controller tuning, ensuring optimal controller performance across various systems. While the study did not adhere to a specific tuning methodology, applying the auto-tuning toolbox secured the selection of appropriate parameters for each system.

A second-order robustness filter $F_r(z)$ was employed in all case studies to accelerate the disturbance rejection. The filter was designed following the procedure outlined in the filter tuning section. Specifically, a second-order filter was selected, with $v = 2$ and $\beta = e^{-2T_s/L_n}$, where L_n is the time delay suggested in NORMEY-RICO *et al.* (2014). The numerator of the filter seeks to eliminate the dominant closed-loop dynamics, λ, finding the second zero as part of the residual polynomial $p_n(z)$. Table 4.2 presents the tuning parameters of the controllers, the sampling time, and the filters used for each case. The tuning parameters for the monitoring algorithm are presented in Table 4.3. The parameter ϕ was selected in such a way as to capture both the transient and the steady state of each case study. The p-band was chosen to contain the process signal and the noise associated with this signal, preventing false positives from the signal moving outside of this band. The parameter α was set to 20% (50% for high order system) variation up and down to find the transfer function parameters in the objective function estimation of Equation (4.1).

Lastly, the horizon of the moving horizon filter, represented as h_s, plays a crucial role in effectively filtering the noise in the system signal. Its selection is based on the noise's nature and the system's dynamic characteristics. Specifically, a more extensive h_s provides a greater degree of noise smoothing, but it might also attenuate sharp changes in the signal caused by sudden disturbances. Conversely, a smaller h_s can capture abrupt changes more quickly, but it might be less effective in filtering out high-frequency noise. Therefore, it's essential to balance these factors and select an h_s that allows for sufficient noise filtering without suppressing the dynamics of interest.

Scenario 1: MPM + UD Detection

The first scenario tests the monitoring and diagnosing of the control structure and self-tuning filter. In this scenario, it is assumed that the monitoring system was initially disabled at the start-up of the control structure. This represents a situation where the monitoring algorithm has not been immediately activated, leading to the presence of MPM in the system from the start of the control structure operation, which deviates from the initial hypothesis considered in the methodology of this

Table 4.2: PI controller parameters, sampling time, and robustness filter of each case study

Case	K_c	τ_i [min]	T_s [min]	$F_r(z)$
First order	15.15	7.2	0.4	$\frac{5.275(z-0.936)(z-0.882)}{(z-0.801)^2}$
Non-minimum phase	0.1142	0.4821	0.4	$\frac{1.371(z-0.924)(z-0.618)}{(z-0.801)^2}$
Oscillatory	0.3774	2.8302	0.4	$\frac{3.721(z-0.932)(z-0.870)}{(z-0.801)^2}$
High order	0.159	5	1	$\frac{0.77(z-0.819)(z-0.652)}{(z-0.779)^2}$
Nonlinear	0.6443	0.6782	0.1	$\frac{2.742(z-0.980)(z-0.946)}{(z-0.801)^2}$

Table 4.3: Tuning parameters for the monitoring algorithm used in each case study

Case	ϕ	p	α	h_s
First order	100	6%	0.2	7
Non-minimum phase	100	6%	0.2	5
Oscillatory	100	8%	0.2	5
High order	100	6%	0.5	7
Nonlinear	150	0.12%	0.2	15

algorithm. The simulation scenario presented here shows how the algorithm behaves when activated in a control structure with MPM presence. Additionally, the auto-tuning of the primary controller is disabled ($sc = OFF$). Therefore, the monitoring algorithm only performs the auto-tuning of the robustness filter and displays alarms in the presence of a possible MPM or UD. Figures 4.5 - 4.9 show the dynamic responses of the control system and the robust stability condition for each case study.

A square pulse generator is applied to the setpoint (varying between 1 and 2 for all systems, but the nonlinear one that was varied between 0 and 0.01 to maintain the system close to the linearization point) every 100 sampling times to create a dynamic response, as shown in subfigures (a). In all cases, at the beginning of the operation, the response is oscillatory and deviates from the p-band. The presence of MPM causes this behavior, and the MPM flag is activated. Despite the presence of MPM, as observed in subfigures (b), the robust stability of the controller is guaranteed for the cases of first-order and non-minimum phase systems, complying with the criterion of Equation (2.62) because the initial robustness index, $dP_i(\omega)$, is greater than the modulus of the domain error shape in the frequency domain, $|\overline{\delta P}(\omega)|$. However, this criterion is not met in the case of high-order, oscillatory and nonlinear systems.

Note that to facilitate comparison and provide a clearer understanding of the performance improvement offered by the proposed monitoring algorithm, the dynamic responses of the control systems with only the PI controller, without the application of the monitoring algorithm, are included. These responses are presented in Appendix C. The comparison between the baseline PI controller responses and the responses obtained with the addition of the monitoring algorithm offers insightful information about the algorithm's effectiveness. It illustrates how introducing the monitoring algorithm and auto-tuning the robust filter can improve the system's behavior, especially in cases where the MPM is present from the start.

The algorithm utilizes the robustness self-tuning filter to reduce the oscillations caused by the MPM during the activation of MPM algorithm detection. In doing so, it adjusts the parameter γ such that the sum of γ and $|\overline{\delta P}(\omega)|$ is always less than $dP(\omega)$, ensuring the robust stability of the system (Equation (4.12)). In all cases, this reduces the final robust stability margin, $dP_f(\omega)$. The sum of γ and $|\overline{\delta P}(\omega)|$ is depicted as the dashed line in the plots, resulting in a conservative dynamic response, oscillating within the p-band, which allows for estimating the transfer function parameters during the next setpoint change. Additionally, it should be noted that an unmeasured abrupt disturbance, $n(k) = 0.5$, is applied in the middle of the first descending setpoint for the first 4 case studies. The nonlinear system's magnitude of the unmeasured abrupt disturbance is $n(k) = 0.003$. The algorithm

promptly activates the UD flag and estimates the gain of this disturbance.

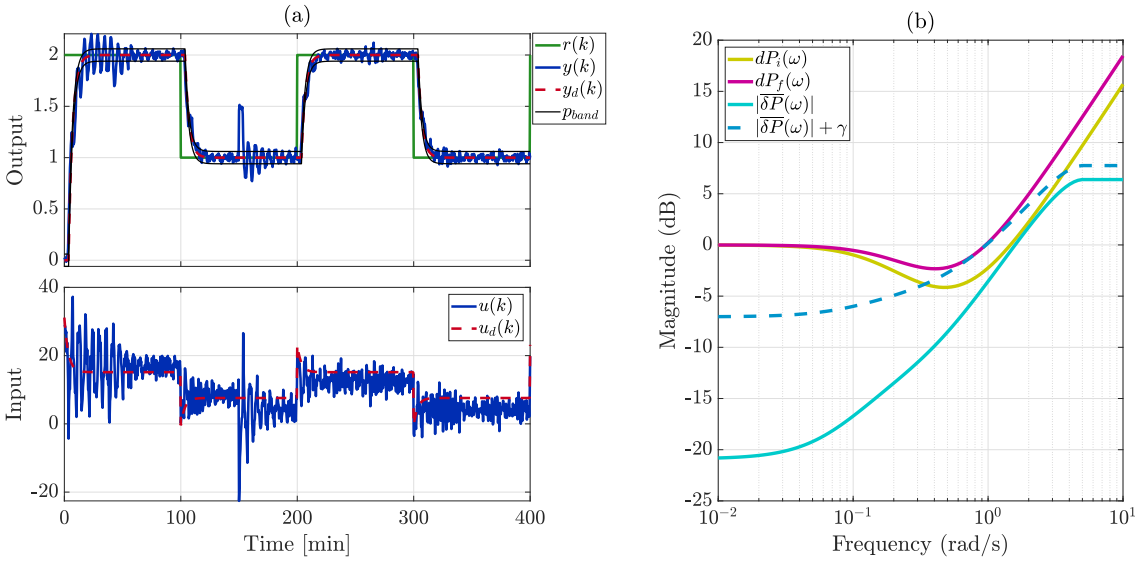


Figure 4.5: First order system: (a) dynamic response; (b) robust stability condition.

This first simulation scenario aims to verify that the proposed algorithm detects the MPM and the presence of an abrupt UD, activating the flags to show the existence or absence of these events in the FSP structure. While monitoring the control structure, this proposal also estimates the internal model parameters. The transfer function parameters are expressed as a continuous-time transfer function with the format presented in Equation (4.16) to generalize the system identification. The estimated parameters of the transfer functions and the MPM flags are summarized in Table 4.4 while the estimation of the magnitude of the abrupt unmeasured disturbances, employing Equation (4.2), and the UD flags are in Table 4.5.

$$P_n(s) = \frac{b_0 s + b_1}{a_0 s^2 + a_1 s + a_2} e^{-L_n s} \tag{4.16}$$

According to Table 4.4, the estimated parameters align well with the nominal model for the first-order system. The time delay L_n is estimated with reasonable accuracy, and the presence of MPM is correctly detected.

The algorithm's performance in estimating the parameters of the non-minimum phase system also has a good agreement. Despite the inherent difficulties of model parameter estimation in the presence of MPM, the estimated parameters approximate well the nominal model parameters. Notably, the time delay L_n is accurately estimated, and the presence of MPM is appropriately identified.

For the oscillatory system, the estimated parameters closely match those of the nominal model. The algorithm successfully identifies the presence of MPM and

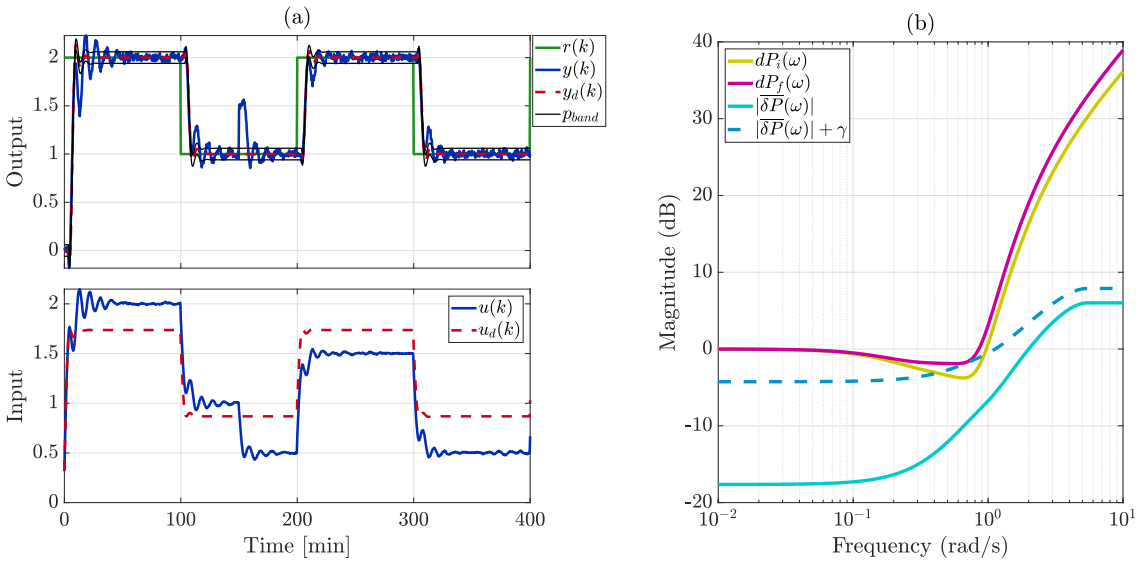


Figure 4.6: Non-minimum system: (a) dynamic response; (b) robust stability condition.

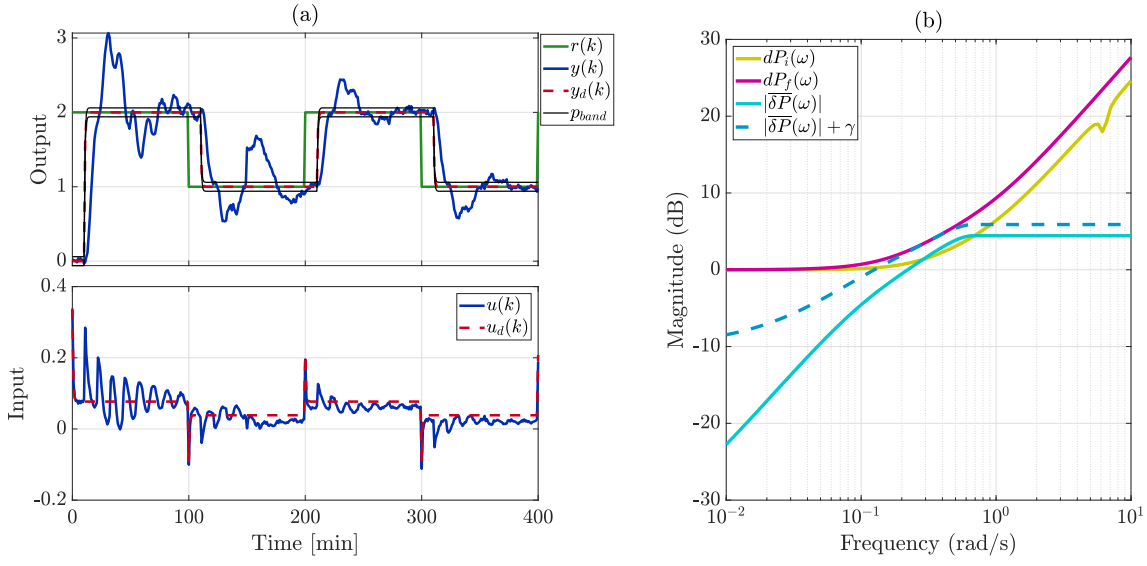


Figure 4.7: High order system: (a) dynamic response; (b) robust stability condition.

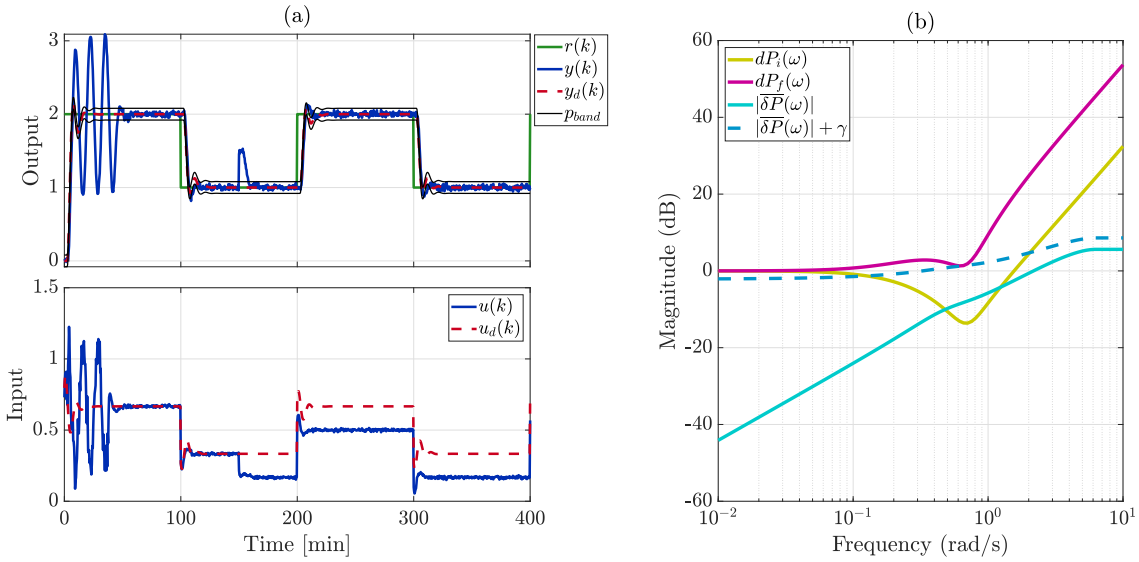


Figure 4.8: Oscillatory system: (a) dynamic response; (b) robust stability condition.

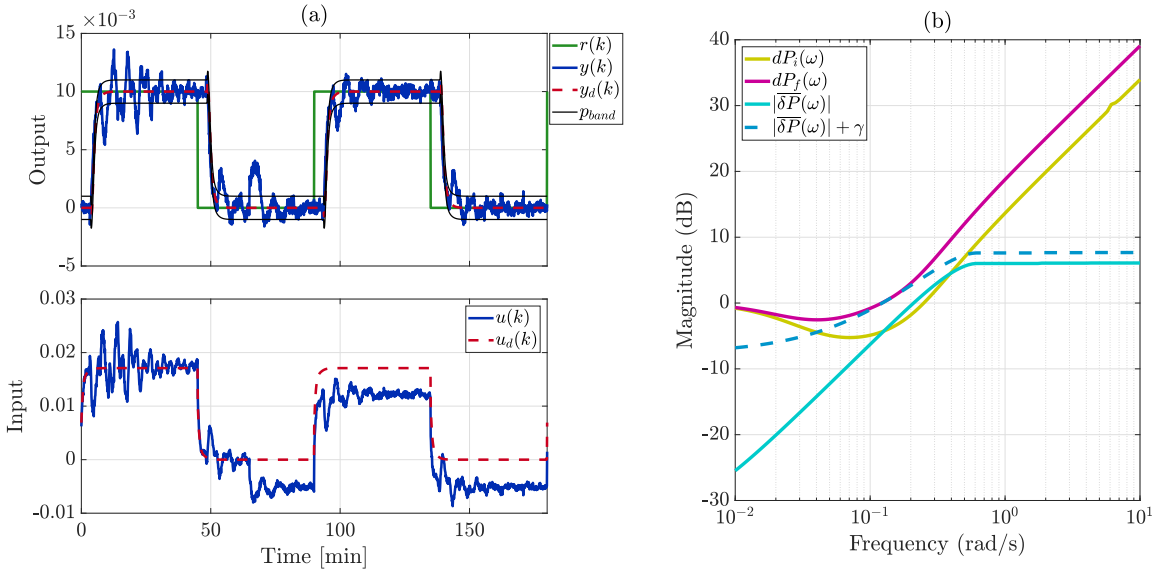


Figure 4.9: Nonlinear system: (a) dynamic response; (b) robust stability condition.

Table 4.4: Estimated parameters of the transfer functions and the MPM flags

System	b_0	b_1	a_0	a_1	a_2	L_n [min]	MPM
First order	0	0.02	0	1	0.166	2.98	1
Non-minimum phase	-1.004	1.002	1	1.003	1.002	3	1
Oscillatory	0	0.75	1	0.601	0.25	3	1
High order	0	3.56	0	1	0.137	14.5	1
Nonlinear	-1.135	3.15	1	4.833	5.515	3.09	1

Table 4.5: Magnitude estimation of the abrupt unmeasured disturbances and the UD flags.

System	τ_n [min]	$n(k)$	UD
First order	0.026	0.456	1
Non-minimum phase	2.79	0.505	1
Oscillatory	0.202	0.498	1
High order	6.2	0.52	1
Nonlinear	0.025	0.0027	1

provides a conservative estimation of the time delay L_n.

In the case of the high-order system, the estimated first-order model presents dynamic characteristics similar to the fifth-order plant. In comparing the two models, emphasis was placed on performance metrics such as the gain, time delay, and time constant rather than the order of the system. Specifically, the estimated parameters yield a model with a gain of 24.67, which closely approximates the gain of the high-order system. Similarly, the time delay L_n in the estimated model is 15.2 [min], which, while slightly greater than that of the high-order system, still provides an adequate representation of the system's dynamic response. Moreover, the time constant of the estimated first-order system is approximately 6 [min], closely mirroring the time constant of the high-order system, which is about 6.25 [min]. This further attests to the comparable dynamics between the two systems.

Finally, the algorithm provides a satisfying estimation of the model parameters for the nonlinear system. The time delay, L_n, is estimated as 3.09 [min], which is quite close to the real plant's time delay of 3.1 [min] and notably less than the original model's time delay of 3.6 [min]. The other parameters of the transfer function are accurately estimated, matching those obtained through linearization of the differential equations of the Van de Vusse reactor system. The presence of MPM, in this case, is solely due to this time delay and not due to any inherent nonlinearity, which underscores the algorithm's robustness in effectively handling complex systems like this.

Scenario 2: UD Detection

The second scenario shows that the algorithm performs well when there is no MPM, and the system is impacted only by measurement noise and abrupt unmeasured disturbances. The higher-order system is selected to verify this second scenario. In this case, the transfer function parameters presented in Table 4.4 are placed in the internal model of the FSP structure. The dynamic response of this case is given in Figure 4.10a. The monitoring algorithm shows that the system does not present MPM and only activates the UD flag, providing an estimate of the magnitude of the disturbance, $n(k) = 0.51$ and $\tau_n = 2.2$ [min]. For this case, the controlled variable's noise is increased 20 times, causing the signal to leave the p-band constantly, so adjusting the p-band for 12%, as shown in Figure 4.10b, it was possible to estimate the magnitude of the UD, $n(k) = 0.51$ and $\tau_n = 0.98$ [min].

4.3.2 Experimental Application

The Arduino-based Temperature Control Laboratory (TCLab) proposed by HEDENGREN (2023); HEDENGREN *et al.* (2019) was used in this work to test

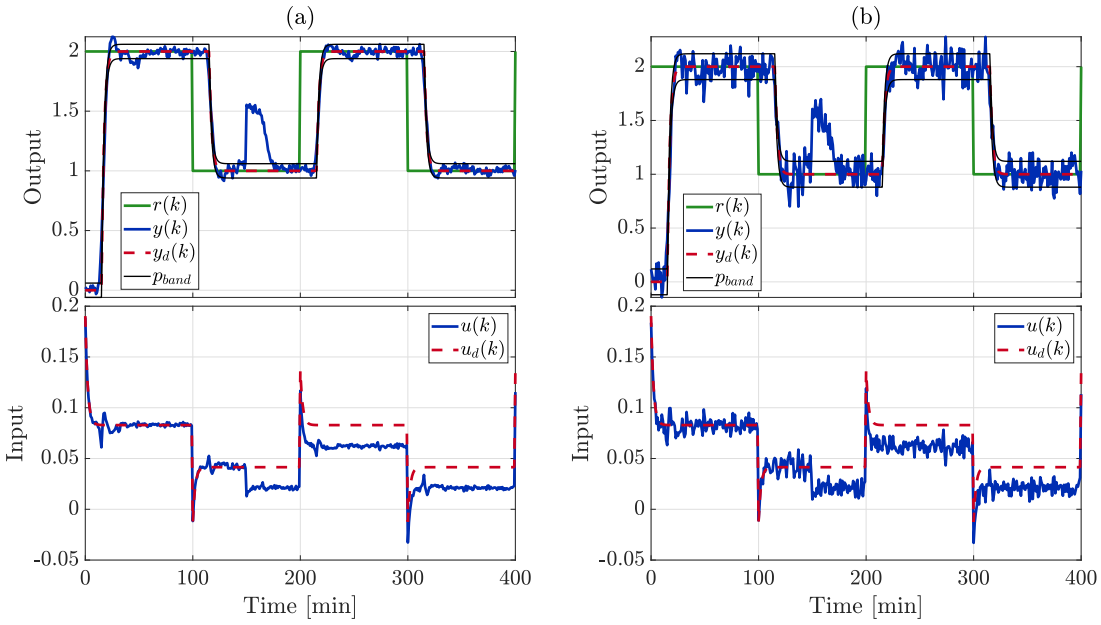


Figure 4.10: High-order system: (a) without MPM; (b) without MPM + higher noise.

the proposed algorithm in an experimental plant. The TCLab, presented in Figure 4.11, has two actuator heaters (NPN Bipolar Junction Transistor), three temperature sensors (thermistor), and two current sensors. This is a modification of the original TCLab, and it is helpful to carry out feedback control practices. Additional information about the TCLab can be found in Appendix E.

For this case study, only one heater and one temperature sensor are considered for the SISO control. The dynamic model between input power to the transistor and the temperature sensed by the thermistor is represented by the following energy balance:

$$mc_p\frac{dT}{dt} = UA(T_\infty - T) + \epsilon\sigma A(T_\infty^4 - T^4) + \alpha Q, \tag{4.17}$$

where $m = 0.004$ [kg] is the mass, $c_p = 500$ [J/kg K] is the heat specific capacity, T is the temperature in Kelvin, $U = 4.1183$ [W/m^2 K] is the heat transfer coefficient, $A = 1.2 \times 10^{-3}$ [m^2] is the heat transfer area, $T_\infty = 298$ [K] is the ambient temperature, $\epsilon = 0.9$ is the emissivity, $\sigma = 5.67 \times 10^{-8}$ [W/m^2 K^4] is the Stefan-Boltzmann constant, and Q is the percentage heater output in Watts. The parameter α is a factor that relates heater output (0-100%) to the power dissipated by the transistor in Watts (HEDENGREN, 2023).

The algorithm for this case study was implemented in Simulink. An artificial time delay of 25 seconds was considered for the thermistor reading in Simulink to increase the time delay. An accurate linear model of the process is presented in Equation (4.18). The good agreement of the model with the temperature control

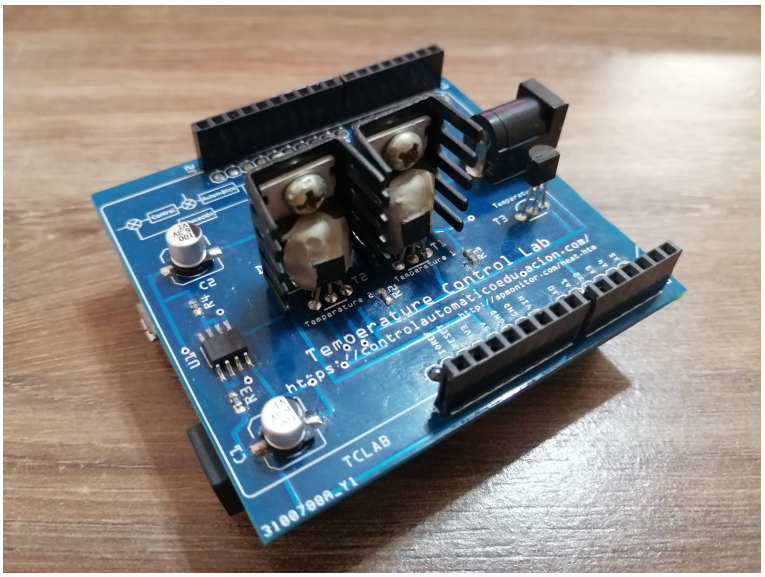


Figure 4.11: TCLab - Temperature Control Lab.

lab is shown through the 33-minutes experiment shown in Figure 4.12, where the average sum of squares error between the model and each observation was 0.30 $[^oC]$.

$$P_n(s) = \frac{1.9386}{151.744s + 1} e^{-44s}. \tag{4.18}$$

A PI controller tuned by the pole assignment method is used as the primary controller of the TCLab. The design criteria were selected for a settling time of 300 [s] and a damping factor of 0.6. Thus, the proportional gain is $K_c = 1.5256$, and the integral time is $\tau_i = 40.35$ [s]. For the $F_r(z)$, it was employed a second-order robustness filter to accelerate the disturbance rejection presented in the following equation:

$$F_r(z) = \frac{0.119z^2 - 4.99z + 1.962}{(z - 0.6967)^2}, \; T_s = 8 \; [s]. \tag{4.19}$$

Note that the initial hypothesis addressed in this work is being respected in this case, where the system starts its operation with an accurate mathematical model. With time, the model will gradually lose its predictive capacity, so the monitoring, diagnosis, and self-tuning system must respond adequately to this event as well as if an unmeasured disturbance is detected. Furthermore, the primary controller's auto-tune feature is enabled ($sc = ON$), allowing the use of any model-based PID controller design strategy from the literature. Therefore, the tuning parameters for the monitoring algorithm were $T_s = 8$ [s], $\phi = 80$, $p = 1.5$, $\alpha = 0.8$, and $h_s = 9$.

Figure 4.13 shows the 83 [min] experiment where the dynamic response of the TCLab can be seen. In the first 500 seconds, the plant response, $y(k)$, is similar to the designed response, $y_d(k)$, due to the minimum MPM in the control structure.

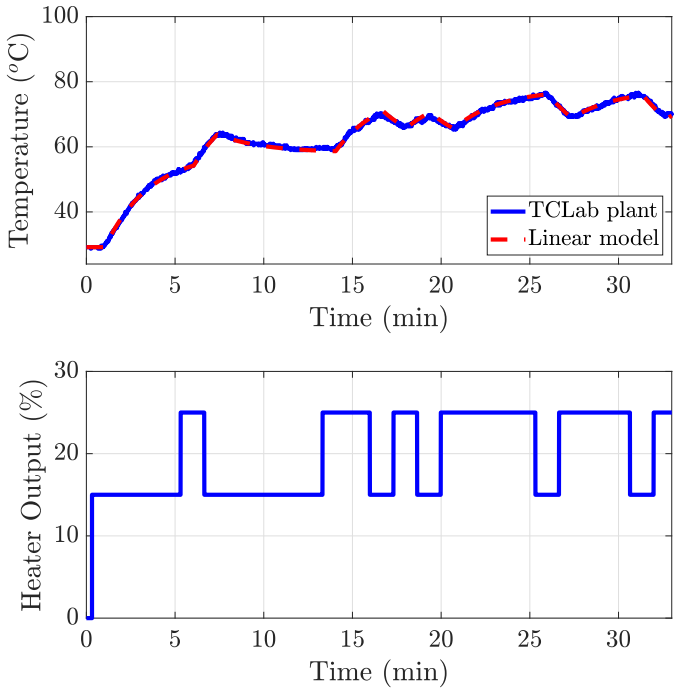


Figure 4.12: Comparison between the measured data and the linear model.

After 500 seconds, a degradation of the real system was simulated. To simulate a decrease in the gain and the time constant in the TCLab plant, the second heater of the system was gradually energized with a ramp with a slope of 0.025 $[\%/s]$ in the time interval between 500 [s] - 900 [s]. This decrease in gain and time constant can be justified as an increase in the power supply voltage of the plant over a certain period. Additionally, an artificial time delay of 8 seconds from 600 seconds was inserted into Simulink, another time delay of 8 seconds from 700 seconds, and finally, a third-time delay of 9 seconds from instant 850 [s]. These artificial time delays can be justified as the delay added by the controller and plant communication system.

At 1,200 seconds, a modification in the setpoint takes place, indicating the onset of an oscillatory reaction caused by MPM. Following the expiration of the ϕ time window, equivalent to 80 sample periods, at 1,840 seconds, the new model estimation begins due to the plant response remaining outside the p-band for a long time. The MPM flag is triggered to warn the user of a model failure. The updated estimated transfer function is presented in Equation (4.20):

$$P_n(s) = \frac{1.66}{128s + 1} e^{-74s}. \tag{4.20}$$

At 1,840 seconds, the self-tuning algorithm for the robustness filter is executed in parallel to reduce the oscillations caused by MPM. Figure 4.14 presents the robust

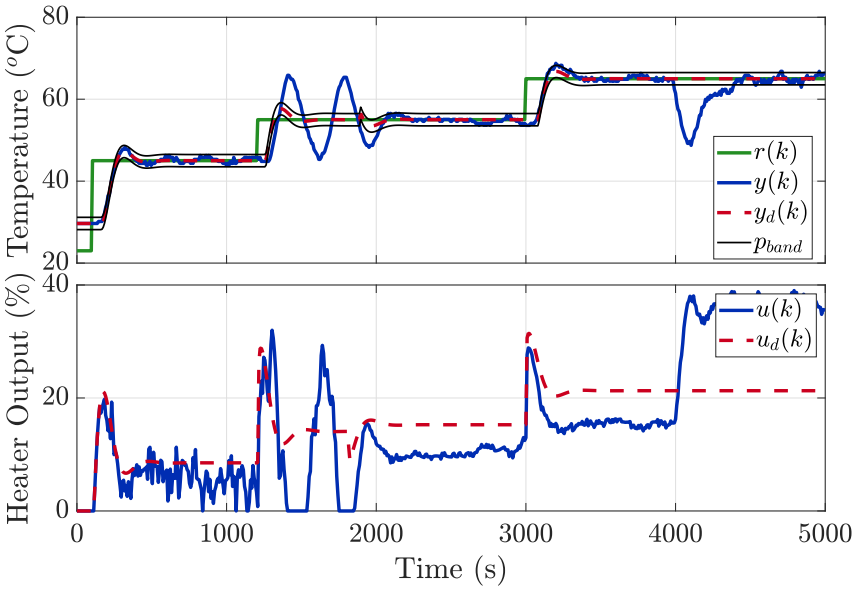


Figure 4.13: Dynamic response of the TCLab.

stability margin of the control structure in the TCLab. The figure illustrates that once the new filter is estimated using $|\overline{\delta P}(\omega)| + \gamma$ as a reference, the final robust stability margin $dP_f(\omega)$ ensures the condition given by Equation (4.12). By implementing this procedure, a conservative dynamic response that oscillates within the p-band can be achieved. The new robustness filter is presented in the following equation:

$$F_r(z) = \frac{0.7298z^2 - 1.177z + 0.4632}{(z - 0.8724)^2}, \; T_s = 8 \; [s]. \tag{4.21}$$

Finally, the new parameters of the PI controller tuned by pole assignment are updated, where $K_c = 1.45$ and $\tau_i = 38.18$ [s].

From Figure 4.13, at 3,000 seconds, a new setpoint change is introduced. It is observed that the behavior of the controlled variable is consistent and similar to the desired response, indicating that the updated tuning of the model, controller, and filter has successfully adapted to the plant's new dynamics. The response always remains within the p-band, so the algorithm does not require new estimates within the control structure. At 4,000 seconds, a sudden and constant disturbance is induced by introducing a flow of cold air, which attempts to lower the temperature of the heaters. The FSP structure effectively rejects this disturbance within 500 seconds. After eliminating this new dynamic, the diagnostic algorithm activates the UD flag, signaling to the user the presence of abrupt disturbances in the control loop. Moreover, the algorithm presents an estimate of the magnitude of the disturbance, given by $n(k) = -13.10$ [^oC] and $\tau_n = 19.5$ [s], in the control panel.

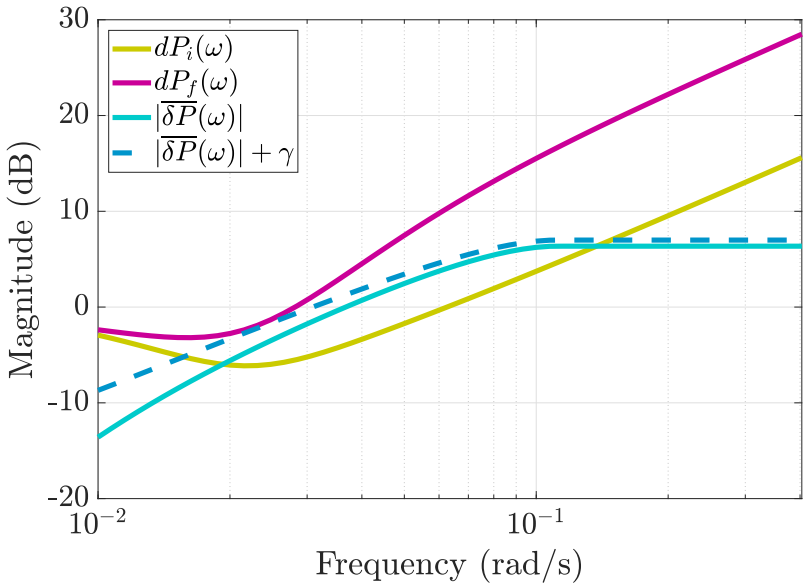


Figure 4.14: Robust stability condition of the TCLab.

The implementation and testing of the proposed algorithm provide valuable insights into its real-time performance and adaptability. The reader is referred to Appendix D for a comparative analysis of the system response when only the primary PI controller is employed, without applying the MPM and UD monitoring system. This comparative analysis is essential to understand the significant improvements in the control system's performance by implementing the proposed algorithm.

4.4 Final Remarks

This chapter has presented a novel algorithm designed for real-time diagnostics in single-input, single-output systems when using dead-time compensators with Filtered Smith Predictors. The unique aspect of this algorithm is its ability to detect model errors and unmeasured disturbances and tune the robustness filter to maintain system stability automatically.

A significant part of the presented approach is the self-tuning robustness filter. The effectiveness of this filter was demonstrated by its ability to produce a smooth and less oscillatory response in the closed-loop control system, thereby ensuring robust stability. The filter and the algorithm's capability to identify and adapt to changes provide an effective solution for high-risk scenarios such as time-delay mismatches.

The validity of the method has been confirmed through several simulated case studies and a real-world temperature control experiment. These tests showed the

method's proficiency in detecting and diagnosing changes in process dynamics and adjusting the controller's parameters to maintain system stability and performance. The simplicity and low computational cost of the algorithm make it a feasible option for real-time applications in any single-input, single-output control system.

For future research, the applicability of the method to multi-input, multi-output systems could be explored. This would further reveal the benefits of the proposed method for designing control systems on a larger scale. By contributing to developing more efficient and reliable control systems, the work presented in this chapter offers a promising path for future advancements in adaptive control.

Chapter 5

Conclusions

In this doctoral research, two significant methodologies have been developed and validated to enhance the performance of model-based predictive controllers. Both the methodologies focus on Model Predictive Control (MPC) and Dead Time Compensator (DTC) systems, but they tackle different aspects of their functioning.

The first part of the research presents an innovative tuning algorithm for MPC controllers. A unique feature of this algorithm is its hybrid nature, as it combines two optimization algorithms to ascertain the best tuning parameters. These parameters include the weighting matrices, prediction, and control horizon. Importantly, the efficacy of this algorithm is not limited to a specific type of system. The versatility of the algorithm is highlighted by its successful application across a range of system types, from linear to non-linear and square to non-square systems.

An essential aspect of the developed algorithm is its resilience against modeling errors or measurement noise. Despite these factors that typically impede the performance of controllers, the algorithm has demonstrated its robustness. Another notable attribute of the algorithm is its adaptability. It can conform to the trajectory defined by the user, offering flexibility in its application. Lastly, the low computational cost of the algorithm adds to its practicality. It enables the efficient finding of MPC tuning parameters without taxing computational resources excessively.

The second half of the thesis sheds light on a novel algorithm developed for real-time diagnostics in single-input, single-output systems, focusing on Dead Time Compensators for Filtered Smith Predictors. This algorithm has proven to be multifunctional. It not only detects model errors and unmeasured disturbances, but it also fine-tunes the robustness filter. In doing so, it guarantees system stability, a critical factor in maintaining the performance of any system.

This algorithm is a significant step in improving industrial process control as it allows real-time monitoring and makes necessary adjustments to ensure the system always operates optimally. This is particularly vital in high-risk scenarios such as time-delay mismatches, where the system's stability can be severely compromised

without the proper corrective measures.

Both the methodologies, with their respective algorithms, have undergone thorough validation through simulated and real-world case studies. These studies have emphasized their practical applicability and effectiveness in enhancing control system performance.

The methodologies and algorithms proposed in this thesis hold promise for broader applications. These could include exploring their functionality in multi-input, multi-output systems and their potential advantages for control system design in large-scale control systems.

In conclusion, this doctoral thesis has achieved its objective of developing and validating methodologies that can effectively enhance the performance of model-based predictive controllers. The proposed approaches and algorithms have demonstrated their potential to improve control system performance and robustness, provide useful real-time diagnostics, and offer adaptability to different system types and user requirements. These achievements hold considerable promise for advancing the field of adaptive control, offering valuable insights and tools to address ongoing challenges in control systems engineering.

Regarding the 'Tuning of Model Predictive Controllers Based on Hybrid Optimization', several avenues for future research present themselves. A valuable next step might involve extending the hybrid optimization approach to include alternative optimization techniques, further fortifying the algorithm's resilience against complex disturbances. Additionally, testing and validating this approach across various domains beyond chemical processes, such as mechanical or electrical systems, would offer broader insights into its adaptability and efficiency. The potential to optimize additional tuning parameters, or to integrate real-time adaptive feedback, might further refine its utility in complex control scenarios

Turning to the 'Filtered Smith Predictor Monitoring, Diagnosis, and Self-tuning due to Unmeasured Abrupt Load Disturbance or Model Plant Mismatch', there exists an exciting potential to enhance real-time system monitoring through the incorporation of more comprehensive fault detection and diagnostic mechanisms. It would be particularly beneficial to extend the algorithm to detect and diagnose disturbances of varying natures, not limited to abrupt disturbances represented by step changes. A significant advancement would be achieved by adapting the algorithm for multi-input, multi-output systems, expanding its practical applications. Further exploration into integrating this method with modern control paradigms, particularly those driven by artificial intelligence, promises innovative solutions to longstanding adaptive control challenges. Pursuing these lines of investigation will undoubtedly propel forward the domain of control systems engineering.

References

ABRASHOV, S., AIRIMITOAIE, T. B., LANUSSE, P., et al., 2017, "Model Predictive Control Tuning: Methods and Issues. Application to steering wheel position control", *IFAC-PapersOnLine*, v. 50, n. 1, pp. 11331–11336. 20th IFAC World Congress.

AL-GHAZZAWI, A., ALI, E., NOUH, A., et al., 2001, "Online Tuning Strategy for Model Predictive Controllers", *Journal of Process Control*, v. 11, n. 3 (06), pp. 265–284.

ALHAJERI, M., SOROUSH, M., 2020, "Tuning Guidelines for Model-Predictive Control", *Industrial & Engineering Chemistry Research*, v. 59, n. 10, pp. 4177–4191.

ALOISE, D. J., ALOISE, D., ROCHA, C. T., et al., 2006, "Scheduling workover rigs for onshore oil production", *Discrete Applied Mathematics*, v. 154, n. 5, pp. 695–702. IV ALIO/EURO Workshop on Applied Combinatorial Optimization.

ÅSTRÖM, K. J., 1970, *Introduction to Stochastic Control Theory*. San Diego, California, Academic Press.

ÅSTRÖM, K. J., WITTENMARK, B., 1990, *Computer-Controlled Systems, Theory and Design (Second ed.)*. Prentice Hall.

BADWE, A. S., GUDI, R. D., PATWARDHAN, R. S., et al., 2009, "Detection of model-plant mismatch in MPC applications", *Journal of Process Control*, v. 19, n. 8, pp. 1305 – 1313.

BADWE, A. S., PATWARDHAN, R. S., SHAH, S. L., et al., 2010, "Quantifying the impact of model-plant mismatch on controller performance", *Journal of Process Control*, v. 20, n. 4, pp. 408 – 425.

BAGHERI, P., KHAKI-SEDIGH, A., 2014, "An analytical tuning approach to multivariable model predictive controllers", *Journal of Process Control*, v. 24, n. 12, pp. 41–54.

BOTELHO, V., TRIERWEILER, J. O., FARENZENA, M., et al., 2015, "Methodology for detecting model-plant mismatches affecting model predictive control performance", *Industrial & Engineering Chemistry Research*, v. 54, n. 48, pp. 12072 – 12085.

BOTELHO, V., TRIERWEILER, J. O., FARENZENA, M., 2016a, "Diagnosis of Poor Performance in Model Predictive Controllers: Unmeasured Disturbance versus Model–Plant Mismatch", *Industrial & Engineering Chemistry Research*, v. 55, n. 44, pp. 11566 – 11582.

BOTELHO, V., TRIERWEILER, J. O., FARENZENA, M., et al., 2016b, "Perspectives and challenges in performance assessment of model predictive control", *The Canadian Journal of Chemical Engineering*, v. 94, n. 7, pp. 1225 – 1241.

BOTELHO, V., TRIERWEILER, J. O., FARENZENA, M., 2021, "MPC model monitoring and diagnosis for non-square systems", *Journal of Process Control*, v. 97, pp. 26–44. ISSN: 0959-1524.

BOYD, S. P., BARRATT, C. H., 1991, *Linear control design: Limits of Performance*. New York, Prentice Hall.

CAMACHO, E. F., BORDONS, C., 2002, *Model predictive control*. London, Springer.

CAMPOS, M., GOMES, M., PEREZ, J., 2013, *Controle Avançado e Otimização na Indústria do Petróleo*. Rio de Janeiro, Interciência.

CHAN, G., HUA, L. X., 2015, "Decentralized robust H∞ model predictive control of discrete-time singular large-scale systems". In: *2015 34th Chinese Control Conference (CCC)*, pp. 4173–4178.

CHEN, L., MA, M., JANG, S.-S., et al., 2009, "Performance assessment of run-to-run control in semiconductor manufacturing based on IMC framework", *International Journal of Production Research*, v. 47, n. 15, pp. 4173 – 4199.

CLARKE, D., MOHTADI, C., TUFFS, P., 1987, "Generalized predictive control—Part I. The basic algorithm", *Automatica*, v. 23, n. 2, pp. 137 – 148.

COETZEE, L. C., CRAIG, I. K., KERRIGAN, E. C., 2010, "Robust Nonlinear Model Predictive Control of a Run-of-Mine Ore Milling Circuit", *IEEE Transactions on Control Systems Technology*, v. 18, n. 1, pp. 222–229.

DE LIMA, R. G., SANTOS, T. L. M., 2015, "Sintonia automatizada do filtro de robustez para o controle de sistemas com atraso (Automated tuning of robustness filter for control systems with delays)." In: *Anais do XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, pp. 498–503, Natal. SBA.

DE SCHUTTER, J., ZANON, M., DIEHL, M., 2020, "TuneMPC—A Tool for Economic Tuning of Tracking (N)MPC Problems", *IEEE Control Systems Letters*, v. 4, n. 4, pp. 910–915.

DEB, K., 2014, "Multi-objective Optimization". In: Burke, E. K., Kendall, G. (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pp. 403–449, Boston, MA, Springer US.

DESBOROUGH, L., HARRIS, T., 1993, "Performance assessment measures for univariate feedforward feedback control", *The Canadian Journal of Chemical Engineering*, v. 71, n. 4, pp. 605 – 616.

DESBOROUGH, L., HARRIS, T., 1992, "Performance assessment measures for univariate feedback control", *The Canadian Journal of Chemical Engineering*, v. 70, n. 6, pp. 1186 – 1197.

DING, S. X., LI, L., 2021, "Control performance monitoring and degradation recovery in automatic control systems: A review, some new results, and future perspectives", *Control Engineering Practice*, v. 111, pp. 104790. ISSN: 0967-0661.

EXADAKTYLOS, V., TAYLOR, C. J., 2010, "Multi-objective performance optimisation for model predictive control by goal attainment", *International Journal of Control*, v. 83, n. 7, pp. 1374–1386.

FLEMING, P., 1986, "Application of Multiobjective Optimization to Compensator Design for SISO Control Systems", *Electronics Letters*, v. 22, n. 5, pp. 258–259.

FLESCH, R. C. C., TORRICO, B. C., NORMEY-RICO, J. E., et al., 2011, "Unified approach for minimal output dead time compensation in MIMO processes", *Journal of Process Control*, v. 21, n. 7, pp. 1080–1091.

FONTES, R. M., MARTINS, M. A. F., ODLOAK, D., 2019, "An Automatic Tuning Method for Model Predictive Control Strategies", *Industrial & Engineering Chemistry Research*, v. 58, n. 47, pp. 21602–21613.

FU, R., XIE, L., SONG, Z., et al., 2012, "PID control performance assessment using iterative convex programming", *Journal of Process Control*, v. 22, n. 9, pp. 1793 – 1799.

GARCÍA, C. A., VELANDIA, J. J., CÉSPEDES, M. A., et al., 2021, "Isobaric Vapor–Liquid Equilibrium for the Binary Mixture of 1-Butanol + Butyl l-Lactate at 1 and 5 kPa", *Journal of Chemical & Engineering Data*, v. 66, n. 5, pp. 2006–2011. doi: 10.1021/acs.jced.0c01068.

GARCÍA, P., ALBERTOS, P., 2013, "Robust tuning of a generalized predictor-based controller for integrating and unstable systems with long time-delay", *Journal of Process Control*, v. 23, n. 8, pp. 1205–1216.

GARRIGA, J. L., SOROUSH, M., 2010, "Model Predictive Control Tuning Methods: A Review", *Industrial & Engineering Chemistry Research*, v. 49, n. 8, pp. 3505–3515.

GEMBICKI, F., 1974, *Vector Optimization for Control with Performance and Parameter Sensitivity Indices*. Ph.d. dissertation, Case Western Reserve University, Cleveland, Ohio.

GIRALDO, S., SUPELANO, R., D'AVILA, T., et al., 2021, "Model predictive control with dead-time compensation applied to a gas compression system", *Journal of Petroleum Science and Engineering*, v. 203, pp. 108580. ISSN: 0920-4105.

GIRALDO, S. A. C., MELO, P. A., SECCHI, A. R., 2019, "Tuning of Model Predictive Control Based on Hybrid Optimization", *IFAC-PapersOnLine*, v. 52, n. 1, pp. 136–141. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.

GIRALDO, S. A. C., FLESCH, R. C. C., NORMEY-RICO, J. E., 2016, "Multivariable Greenhouse Control Using the Filtered Smith Predictor", *J. Control, Autom. and Electr. Syst.*, pp. 1–10.

GIRALDO, S. A., MELO, P. A., SECCHI, A. R., 2022, "Tuning of Model Predictive Controllers Based on Hybrid Optimization†", *Processes*, v. 10, n. 2. ISSN: 22279717. doi: 10.3390/pr10020351.

GRAICHEN, K., HAGENMEYER, V., ZEITZ, M., 2004, "Van de Vusse CSTR as a benchmark problem for nonlinear feedforward control design techniques", *IFAC Proceedings Volumes*, v. 37, n. 13, pp. 1123 – 1128. 6th IFAC Symposium on Nonlinear Control Systems 2004 (NOLCOS 2004), Stuttgart, Germany, 1-3 September, 2004.

HAN, K., ZHAO, J., QIAN, J., 2006, "A Novel Robust Tuning Strategy for Model Predictive Control". In: *2006 6th World Congress on Intelligent Control and Automation*, v. 2, pp. 6406–6410.

HANSEN, P., MLADENOVIĆ, N., UROŠEVIĆ, D., 2006, "Variable neighborhood search and local branching", *Computers & Operations Research*, v. 33, n. 10, pp. 3034–3045. Part Special Issue: Constraint Programming.

HANSEN, P., MLADENOVIĆ, N., MORENO PÉREZ, J. A., 2010, "Variable neighbourhood search: methods and applications", *Annals of Operations Research*, v. 175, n. 1, pp. 367–407.

HARRIS, T. J., 1989, "Assessment of control loop performance", *The Canadian Journal of Chemical Engineering*, v. 67, n. 5, pp. 856 – 861.

HARRIS, T., BOUDREAU, F., MACGREGOR, J., 1996, "Performance assessment of multivariable feedback controllers", *Automatica*, v. 32, n. 11, pp. 1505 – 1518.

HARRIS, T., SEPPALA, C., DESBOROUGH, L., 1999, "A review of performance monitoring and assessment techniques for univariate and multivariate control systems", *Journal of Process Control*, v. 9, n. 1, pp. 1 – 17.

HARRISON, C. A., QIN, S. J., 2009, "Discriminating between disturbance and process model mismatch in model predictive control", *Journal of Process Control*, v. 19, n. 10, pp. 1610 – 1616.

HEDENGREN, J. D., 2023. "Temperature Control Lab Kit". https://apmonitor.com/heat.htm. Accessed: 2023-01-12.

HEDENGREN, J. D., MARTIN, R. A., KANTOR, J., et al., 2019, "Temperature Control Lab for Dynamics and Control". In: *AIChE Annual Meeting*, Orlando, FL.

HUANG, B., KADALI, R., 2008, *Dynamic Modeling, Predictive Control and Performance Monitoring*. London, Springer.

HUANG, B., SHAH, S. L., 1999, *Performance assessment of control loops*. Berlin, Springer.

JELALI, M., 2013, *Control Performance Management in Industrial Automation: Assessment, Diagnosis and Improvement of Control Loop Performance*. Advances in Industrial Control. 1 ed. , Springer-Verlag London.

JULIEN, R. H., FOLEY, M. W., CLUETT, W. R., 2004, "Performance assessment using a model predictive control benchmark", *Journal of Process Control*, v. 14, n. 4, pp. 441 – 456.

KO, B., EDGAR, T. F., 2001, "Performance assessment of constrained model predictive control systems", *AIChE Journal*, v. 47, n. 6 (6), pp. 1363 – 1371.

KWAKERNAAK, H., SIVAN, R., 1972, *Linear optimal control systems*. New York, Wiley.

LEE, K. H., HUANG, B., TAMAYO, E. C., 2008, "Sensitivity analysis for selective constraint and variability tuning in performance assessment of industrial MPC", *Control Engineering Practice*, v. 16, n. 10, pp. 1195 – 1215.

LIMA, F., GEORGAKIS, C., 2006, "OPERABILITY OF MULTIVARIABLE NON-SQUARE SYSTEMS", *IFAC Proceedings Volumes*, v. 39, n. 2, pp. 989–994. 6th IFAC Symposium on Advanced Control of Chemical Processes.

LING, D., ZHENG, Y., ZHANG, H., et al., 2017, "Detection of model-plant mismatch in closed-loop control system", *Journal of Process Control*, v. 57, n. Supplement C, pp. 66 – 79.

LJUNG, L., 1999, *System Identification: Theory for the User*. Englewood Cliffs, Prentice-Hall.

LOZANO SANTAMARÍA, F., GÓMEZ, J. M., 2016, "An Algorithm for Tuning NMPC Controllers with Application to Chemical Processes", *Industrial & Engineering Chemistry Research*, v. 55, n. 34, pp. 9215–9228.

MACIEJOWSKI, J. M., 2002, *Predictive control: with Constraints*. Prentice Hall.

MLADENOVIĆ, N., HANSEN, P., 1997, "Variable neighborhood search", *Computers & Operations Research*, v. 24, n. 11, pp. 1097–1100.

NERY JÚNIOR, G. A., MARTINS, M. A., KALID, R., 2014, "A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty", *ISA Transactions*, v. 53, n. 2, pp. 560–567. ISSN: 0019-0578. doi: https://doi.org/10.1016/j.isatra.2013.12.019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0019057813002292>.

NORMEY-RICO, J. E., CAMACHO, E. F., 2007, *Control of dead-time processes*. London, Springer.

NORMEY-RICO, J. E., CAMACHO, E. F., 2008, "Dead-time compensators: a survey", *Control Engineering Practice*, v. 16, n. 4, pp. 407–428.

NORMEY-RICO, J. E., CAMACHO, E. F., 2009, "Unified approach for robust dead-time compensator design", *J. Process Control*, v. 19, n. 1, pp. 38–47.

NORMEY-RICO, J. E., SARTORI, R., VERONESI, M., et al., 2014, "An automatic tuning methodology for a unified dead-time compensator", *Control Engineering Practice*, v. 27, pp. 11–22.

PALMOR, Z. J., 1996, "Time-delay Compensation - Smith Predictor and its Modifications". cap. 10.8, pp. 224–237, CRC and IEEE Press, Boca Raton.

POUR, N. D., HUANG, B., SHAH, S., 2010, "Performance assessment of advanced supervisory–regulatory control systems with subspace LQG benchmark", *Automatica*, v. 46, n. 8, pp. 1363 – 1368.

QIN, S., BADGWELL, T. A., 2003, "A survey of industrial model predictive control technology", *Control Engineering Practice*, v. 11, n. 7, pp. 733–764.

RANI, K., UNBEHAUEN, H., 1997, "Study of predictive controller tuning methods", *Automatica*, v. 33, n. 12, pp. 2243–2248. ISSN: 0005-1098.

REYNOSO-MEZA, G., GARCIA-NIETO, S., SANCHIS, J., et al., 2013, "Controller Tuning by Means of Multi-Objective Optimization Algorithms: A Global Tuning Framework", *IEEE Transactions on Control Systems Technology*, v. 21, n. 2, pp. 445–458.

RICKER, N. L., 1985, "Use of quadratic programming for constrained internal model control", *Industrial & Engineering Chemistry Process Design and Development*, v. 24, n. 4, pp. 925–936.

SANTOS, J. E. W., TRIERWEILER, J. O., FARENZENA, M., 2017, "Model Predictive Control Tuning Strategy for Non-Square Systems and Range Controlled Variables Based on Multi-Scenarios Approach", *Industrial & Engineering Chemistry Research*, v. 56, n. 40, pp. 11496–11506.

SCOKAERT, P. O. M., MAYNE, D. Q., 1998, "Min-max feedback model predictive control for constrained linear systems", *IEEE Transactions on Automatic Control*, v. 43, n. 8 (Aug), pp. 1136–1142.

SHEIKHI, M. A., KHAKI-SEDIGH, A., NIKOOFARD, A., 2021, "Design of nonlinear predictive generalized minimum variance control for performance monitoring of nonlinear control systems", *Journal of Process Control*, v. 106, pp. 54–71. ISSN: 0959-1524.

SHINSKEY, F., 1996, *Process-control systems: application, design, and tuning*. Advances in Industrial Control. New York, NY, USA, McGraw Hill.

SHRIDHAR, R., COOPER, D. J., 1997, "A Tuning Strategy for Unconstrained SISO Model Predictive Control", *Industrial & Engineering Chemistry Research*, v. 36, n. 3, pp. 729–746.

SHRIDHAR, R., COOPER, D. J., 1998, "A Tuning Strategy for Unconstrained Multivariable Model Predictive Control", *Industrial & Engineering Chemistry Research*, v. 37, n. 10, pp. 4003–4016.

SMITH, O. J. M., 1957, "Closer Control of Loops with Dead Time", *Chemical Engineering Progress*, v. 53, n. 5, pp. 217–219.

SUN, Z., QIN, S. J., SINGHAL, A., et al., 2013, "Performance monitoring of model-predictive controllers via model residual assessment", *Journal of Process Control*, v. 23, n. 4, pp. 473 – 482.

TRAN, Q. N., ÖZKAN, L., BACKX, A. C. P. M., 2015, "Generalized predictive control tuning by controller matching", *Journal of Process Control*, v. 25, pp. 1–18.

TRIERWEILER, J. O., 1997, *A Systematic Approach to Control Structure Design*. phd. thesis, Universität Dortmund, Germany, Dortmund.

TRIERWEILER, J., FARINA, L., 2003, "RPN tuning strategy for model predictive control", *Journal of Process Control*, v. 13, n. 7, pp. 591–598.

VALLERIO, M., IMPE, J. V., LOGIST, F., 2014, "Tuning of NMPC controllers via multi-objective optimisation", *Computers & Chemical Engineering*, v. 61, pp. 38–50.

VAN DER LAAN, M. D., 1994, "Robust Sampling for Process Control". In: Halang, W. A., Stoyenko, A. D. (Eds.), *Real Time Computing*, pp. 698–700, Berlin, Heidelberg. Springer Berlin Heidelberg.

VAN DER LEE, J., SVRCEK, W., YOUNG, B., 2008, "A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making", *ISA Transactions*, v. 47, n. 1, pp. 53–59.

VELANDIA, J. J., GARCÍA, C. A., CÉSPEDES, M. A., et al., 2021, "Reactive and non-reactive residue curve maps analysis to produce Butyl Lactate by catalytic distillation", *Chemical Engineering and Processing - Process Intensification*, v. 168, pp. 108558. ISSN: 0255-2701. doi: https:

//doi.org/10.1016/j.cep.2021.108558. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0255270121002531>.

VERONESI, M., VISIOLI, A., 2008, "A Technique for Abrupt Load Disturbance Detection in Process Control Systems", *IFAC Proceedings Volumes*, v. 41, n. 2, pp. 14900–14905. ISSN: 1474-6670. 17th IFAC World Congress.

WEI LIU, GEORGE WANG, 2000, "Auto-tuning procedure for model-based predictive controller". In: *SMC 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. "cybernetics evolving to systems, humans, organizations, and their complex interactions" (cat. no.0*, v. 5, pp. 3421–3426 vol.5.

WELLSTEAD, P. E., ZARROP, M. B., 1991, *Self-Tuning Systems: Control and Signal Processing*. 1st ed. New York, NY, USA, John Wiley & Sons, Inc.

WOOD, R., BERRY, M., 1973, "Terminal composition control of a binary distillation column", *Chemical Engineering Science*, v. 28, n. 9, pp. 1707–1717. ISSN: 0009-2509.

YAMASHITA, A. S., ZANIN, A. C., ODLOAK, D., 2016, "Tuning of Model Predictive Control with Multi-Objective Optimization", *Brazilian Journal of Chemical Engineering*, v. 33, pp. 333–346.

YERRAMILLI, S., TANGIRALA, A. K., 2016, "Detection and diagnosis of model-plant mismatch in MIMO systems using plant-model ratio", *IFAC-PapersOnLine*, v. 49, n. 1, pp. 266 – 271. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.

ZHONG, Q., 2006, *Robust control of time-delay system*. London, Springer.

Appendix A

Adequacy of the Process Model to Be Controlled

Selection of the sampling time (T_s) is performed in the experiment design stage and is a very important step for modeling. This parameter has a direct impact on the tuning procedure and on the MPC controller performance. A short sampling time will unnecessarily overload the processor. This generates an MPC with a high computational cost because p and $\mathbf{m}$ will be elevated. On the other side, a long sampling time will miss the dynamics of the plant to be controlled, generating an MPC with poor or even unstable performance. According to the Nyquist criterion, the sampling frequency should be at least twice as high as the bandwidth of the error signal. This bandwidth is bounded by the system bandwidth. However, to guarantee a satisfactory response, a factor of 10 to 20 may be required VAN DER LAAN (1994). Therefore, a rule of thumb is to sample at least 10 times faster than the settling time of the plant.

Once the system model is obtained, it is important to mention that multivariable systems may have ill-conditioned matrices. Therefore, the MPCT algorithm scales the internal model of the controller to avoid this problem.

For instance, when a linear MPC is implemented, it is possible to use the following optimization problem to scale the system model by minimizing the conditioning number of $\mathbf{P_n}(z)$ (TRIERWEILER and FARINA, 2003), where $\mathbf{P_n}(z)$ represents the linear model of the system, $\mathbf{y}(z) = \mathbf{P_n}(z)\mathbf{u}(z)$:

$$\min_{\mathbf{L},\mathbf{R}} \beta[\mathbf{L}\mathbf{P_n}(z)\mathbf{R}], \tag{A.1}$$

where $\mathbf{L}$ and $\mathbf{R}$ are diagonal matrices and $\beta[\mathbf{P_n}]$ is the conditioning number of matrix $\mathbf{P_n}$. The scaled model, $\mathbf{y}_s(z) = \mathbf{L}\mathbf{y}(z)$ and $\mathbf{u}_s(z) = \mathbf{R}^{-1}\mathbf{u}(z)$, obtained from Equation (A.1), is used for simulation and controller design. The MATLAB function for obtaining the minimum conditioning number for linear processes (both

square and non-square systems) can be found in the following GitHub repository: CondMin.m.

Following the scaling of the linear system model, it's paramount to apply the same scaling to other controller parameters to maintain consistency. Specifically, these include the constraints of the manipulated variables, $\mathbf{u_{min}}$ and $\mathbf{u_{max}}$, the increments of the manipulated variables, $\mathbf{\Delta u_{min}}$ and $\mathbf{\Delta u_{max}}$, the measured disturbances, and the nominal input values of the model. These should be pre-multiplied by $\mathbf{R}^{-1}$. On the other hand, the controlled variables, $\mathbf{y_{min}}$ and $\mathbf{y_{max}}$, the nominal output values of the model, and the scale factors should be pre-multiplied by $\mathbf{L}$.

This additional scaling step is vital as it ensures uniformity across all the parameters involved, thereby enhancing the overall controller performance and stability. Without this crucial scaling step, mismatches in variable magnitudes could lead to suboptimal controller behavior and, in extreme cases, might result in ill-conditioned matrix problems.

For a nonlinear MPC (NMPC) case, it is important to scale the model due to the orders of magnitude between the different MIMO system variables, i.e., temperature, level, pressure, etc. Scaling the model allows one to quantify the margin of error of the variables in relation to references. In this paper, the input and output variables of the model are scaled between 0 and 1 using the following equation:

$$z_s = \frac{z - z_{min}}{z_{max} - z_{min}}, \tag{A.2}$$

where z_s is the scaled variable, z is the original variable, and $[z_{min}, z_{max}]$ is the interval between the minimum and the maximum variable value.

Appendix B

DTC-GPC Implementation Example

The Dead Time Compensator Generalized Predictive Control (DTC-GPC) is applied to the water-methanol distillation column, originally modeled by WOOD and BERRY (1973). This process is a MIMO (Multiple Input Multiple Output) system with strong interactions and inherent time delays. The transfer function matrix of the system is represented as follows:

$$\mathbf{P}(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}, \ \mathbf{P_q}(s) = \begin{bmatrix} \dfrac{3.8e^{-8.1s}}{14.9s+1} \\ \dfrac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix}. \tag{B.1}$$

The MATLAB code implementation of the DTC-GPC for this example can be found in the GitHub repository: https://github.com/sergioacg/Model-Predictive-Control/tree/main/DTC-GPC

Here, the relationship is defined as $\mathbf{y}(s) = \mathbf{P}(s)\mathbf{u}(s) + \mathbf{P_q}(s)q(s)$, where $\mathbf{y}(s) = [y_1(s), y_2(s)]^T$ and $\mathbf{u}(s) = [u_1(s), u_2(s)]^T$. The variables $y_1(s)$ and $y_2(s)$ represent the overhead and bottom product mole fractions of methanol, respectively, and $q(s)$ is the feed flow rate. Meanwhile, $u_1(s)$ and $u_2(s)$ are the reflux and reboiler steam flow rates, respectively. The time delays and constants are in units of minutes, mole fractions are expressed as a percentage, and the flow rates are in units of pounds per minute.

First, it is necessary to scale the process model using the following diagonal matrices: $\mathbf{L} = \text{diag}[0.4103, 0.5640]$ and $\mathbf{R} = \text{diag}[0.6120, 0.2937]$ found through the solution of Equation (A.1). The scaled model process is given by:

$$\mathbf{P_n}(s) = \begin{bmatrix} \dfrac{3.214e^{-s}}{16.7s+1} & \dfrac{-2.278e^{-2s}}{21s+1} \\ \dfrac{2.278e^{-2s}}{10.9s+1} & \dfrac{-3.214e^{-s}}{14.4s+1} \end{bmatrix}. \tag{B.2}$$

$\mathbf{P_n}(z^{-1})$ is obtained from a zero-order hold discretization of $\mathbf{P_n}(s)$ with a sampling period of $T_s = 1$ min, and is given by:

$$\mathbf{P_n}(z^{-1}) = \begin{bmatrix} \dfrac{0.1868z^{-1}}{1-0.9419z^{-1}}z^{-1} & \dfrac{-0.1059z^{-1}}{1-0.9535z^{-1}}z^{-2} \\ \dfrac{0.1997z^{-1}}{1-0.9123z^{-1}}z^{-2} & \dfrac{-0.2156z^{-1}}{1-0.9329z^{-1}}z^{-1} \end{bmatrix}, \tag{B.3}$$

where $\mathbf{y}(k) = \mathbf{P_n}(z^{-1})\mathbf{u}(k)$. The effective dead time of the i output is $d_1 = 1$ and $d_2 = 1$, which allows the MIMO model to be described as $\mathbf{y}(k) = \mathbf{L}(z^{-1})\mathbf{G_n}(z^{-1})z^{-1}\mathbf{u}(k)$ or simply $\mathbf{y}(k) = \mathbf{L}(z^{-1})\mathbf{G_n}(z^{-1})\mathbf{u}(k-1)$, so the output prediction of the model can be calculated with:

$$\mathbf{y}(k) = \begin{bmatrix} z^{-1} & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} \dfrac{0.1868}{1-0.9419z^{-1}} & \dfrac{-0.1059}{1-0.9535z^{-1}}z^{-1} \\ \dfrac{0.1997}{1-0.9123z^{-1}}z^{-1} & \dfrac{-0.2156}{1-0.9329z^{-1}} \end{bmatrix} \mathbf{u}(k-1). \tag{B.4}$$

The least common multiples of the denominators are:

$$\begin{aligned} A_1(z^{-1}) &= 1 - 1.8954z^{-1} + 0.8981z^{-2} \\ A_2(z^{-1}) &= 1 - 1.8453z^{-1} + 0.8511z^{-2} \end{aligned} \tag{B.5}$$

The corresponding numerators are:

$$\begin{aligned} B_{11}(z^{-1}) &= 0.1868 - 0.1781z^{-1}, \quad B_{12}(z^{-1}) = -0.1059z^{-1} + 0.0998z^{-2} \\ B_{21}(z^{-1}) &= 0.1997z^{-1} - 0.1863z^{-2}, \quad B_{22}(z^{-1}) = -0.2156 + 0.1967z^{-1}. \end{aligned} \tag{B.6}$$

The output prediction of the process is computed using the diophantine equation and considers a prediction horizon as $p = 3$ and a control horizon as $m = 3$. So, the

output prediction is expressed as Equation (2.73), which in its expanded form is:

$$\begin{bmatrix} \hat{y}_1(k+2|k) \\ \hat{y}_1(k+3|k) \\ \hat{y}_1(k+4|k) \\ \hat{y}_2(k+2|k) \\ \hat{y}_2(k+3|k) \\ \hat{y}_2(k+4|k) \end{bmatrix} = \begin{bmatrix} 0.1868 & 0 & 0 & 0 & 0 & 0 \\ 0.3628 & 0.1868 & 0 & -0.1059 & 0 & 0 \\ 0.5285 & 0.3628 & 0.1868 & -0.2069 & -0.1059 & 0 \\ 0 & 0 & 0 & -0.2156 & 0 & 0 \\ 0.1997 & 0 & 0 & -0.4168 & -0.2156 & 0 \\ 0.3819 & 0.1997 & 0 & -0.6045 & -0.4168 & -0.2156 \end{bmatrix} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \Delta u_1(k+2) \\ \Delta u_2(k) \\ \Delta u_2(k+1) \\ \Delta u_2(k+2) \end{bmatrix}$$

$$+ \begin{bmatrix} -0.1781z^{-1} & -0.1059z^{-1} + 0.0998z^{-2} \\ -0.5157z^{-1} & -0.2069z^{-1} + 0.2889z^{-2} \\ -0.9957z^{-1} & -0.3033z^{-1} + 0.5577z^{-2} \\ 0.1997z^{-1} - 0.1863z^{-2} & 0.1967z^{-1} \\ 0.3819z^{-1} - 0.5301z^{-2} & 0.5597z^{-1} \\ 0.5481z^{-1} - 1.0058z^{-2} & 1.0622z^{-1} \end{bmatrix} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \end{bmatrix}$$

$$+ \begin{bmatrix} 2.8954z^{-1} - 2.7935z^{-2} + 0.8981z^{-3} & 0 \\ 5.5897z^{-1} - 7.1900z^{-2} + 2.6003z^{-3} & 0 \\ 8.9944z^{-1} - 13.0144z^{-2} + 5.0200z^{-3} & 0 \\ 0 & 2.8453z^{-1} - 2.6964z^{-2} + 0.8511z^{-3} \\ 0 & 5.3991z^{-1} - 6.8208z^{-2} + 2.4217z^{-3} \\ 0 & 8.5410z^{-1} - 12.1363z^{-2} + 4.5953z^{-3} \end{bmatrix} \begin{bmatrix} \hat{y}_1(k+1|k) \\ \hat{y}_2(k+1|k) \end{bmatrix}. \tag{B.7}$$

The following step is to calculate the matrix $\mathbf{K}$ with $\mathbf{Q} = \mathbf{W} = \mathbf{I}$:

$$\mathbf{K} = \begin{bmatrix} 0.1321 & 0.2106 & 0.2662 & -0.0339 & 0.0577 & 0.1040 \\ -0.0293 & 0.1028 & 0.2158 & -0.0136 & -0.0651 & 0.0623 \\ -0.0093 & -0.0260 & 0.1401 & -0.0009 & -0.0117 & -0.0284 \\ 0.0293 & -0.0058 & -0.0216 & -0.1517 & -0.2265 & -0.2716 \\ 0.0157 & 0.0544 & -0.0027 & 0.0354 & -0.1056 & -0.2203 \\ 0.0042 & 0.0169 & 0.0327 & 0.0126 & 0.0391 & -0.1425 \end{bmatrix}, \tag{B.8}$$

where only the first m rows of $\mathbf{K}$ (defined as $\mathbf{K}_m$) have to be computed:

$$\mathbf{K}_m = \begin{bmatrix} 0.1321 & 0.2106 & 0.2662 & -0.0339 & 0.0577 & 0.1040 \\ 0.0293 & -0.0058 & -0.0216 & -0.1517 & -0.2265 & -0.2716 \end{bmatrix}. \tag{B.9}$$

The $\mathbf{F_r}(z)$ filter is specifically designed to effectively reject disturbances over a time span of around 15 minutes. This is achieved by canceling out all the open-loop

poles of the plant model that reside outside a circle with a radius of $e^{-(3/15)Ts} \approx 0.8$ through the utilization of the predictor filter (Equation (4.9)). The selection of the filter's poles deliberately aims for slightly faster dynamics compared to the desired response, represented by $\alpha_1 = \alpha_2 = 0.7$. Consequently, the predictor filter can be represented as follows:

$$\mathbf{F_r}(z^{-1}) = \begin{bmatrix} \dfrac{1.495 - 2.303z^{-1} + 0.8981z^{-2}}{(1 - 0.7z^{-1})^2} & 0 \\ 0 & \dfrac{1.445 - 2.206zz^{-1} + 0.8511z^{-2}}{(1 - 0.7z^{-1})^2} \end{bmatrix}. \tag{B.10}$$

For the simulation, the reference of the overhead mole fraction (y_1) is increased by 0.8 mol/mol at $t = 10$ min, the reference of the bottom product mole fraction (y_2) is increased by 0.5 mol/mol at $t = 60$ min, and the feed flow rate (q) is decreased by 0.25 lb/min at $t = 140$ min. The references are plotted in dotted lines. Simulation results are presented in Figure B.1 and show how the use of the DTC-GPC can improve the disturbance rejection response. In the nominal case, the set-point tracking responses of DTC-GPC are fine-tuned to avoid pole-zero cancellations, thus the disturbance rejection response can be faster than the open-loop dynamics of $\mathbf{P_n}(z^{-1})$.

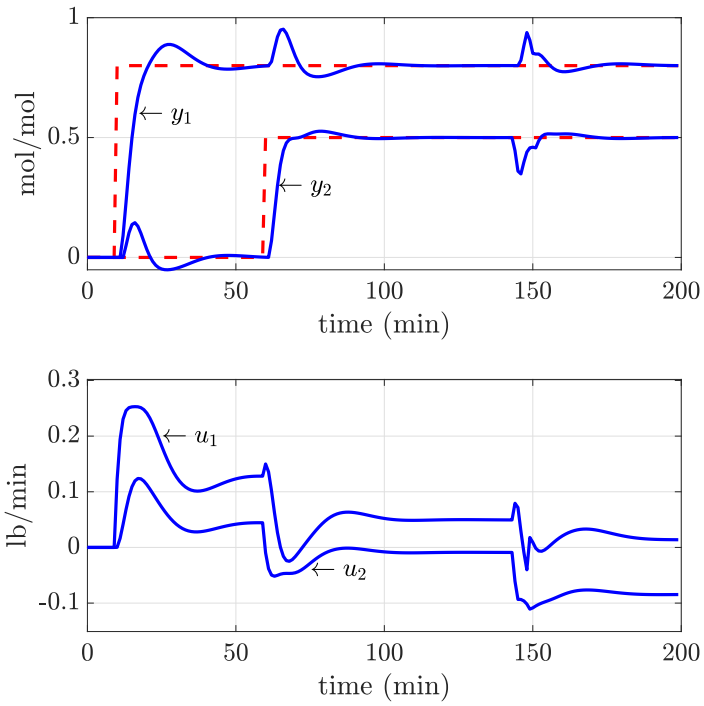


Figure B.1: Behavior of the DTC-GPC.

Appendix C

Comparative Performance Evaluation of PI Control With and Without MPM and UD Monitoring Structure

This appendix presents the dynamic responses of various control systems operating with only a Proportional-Integral (PI) controller without applying the proposed monitoring algorithm presented in Chapter 4. This provides a baseline for the performance comparison of the systems with and without the monitoring algorithm, which is discussed in detail in the main body of the paper.

Figure C.1 depicts the response of the first-order system. Although presenting poor performance, this system maintains stability without applying the monitoring algorithm. Similarly, the non-minimum phase system, portrayed in Figure C.2, remains stable without the monitoring algorithm.

However, the high-order, oscillatory, and nonlinear systems do not exhibit the same stability. Presented in Figures C.3, C.4, and C.5, respectively, these systems become unstable without the implementation of the monitoring algorithm, which underscores the negative impact of the MPM and UD on the system performance if not adequately addressed.

The comparative evaluation between these responses and those provided in the main body of the work, where the monitoring algorithm is implemented, unequivocally highlights the effectiveness of the proposed methodology in counteracting the effects of MPM and enhancing the overall stability and performance of the system.

Table C.1 reports the performance metrics for the PI control with the Monitoring Structure. The metrics include the Integral of the Squared Error (ISE), the Integral of the Time-weighted Squared Error (ITSE), the Integral of the Absolute Error

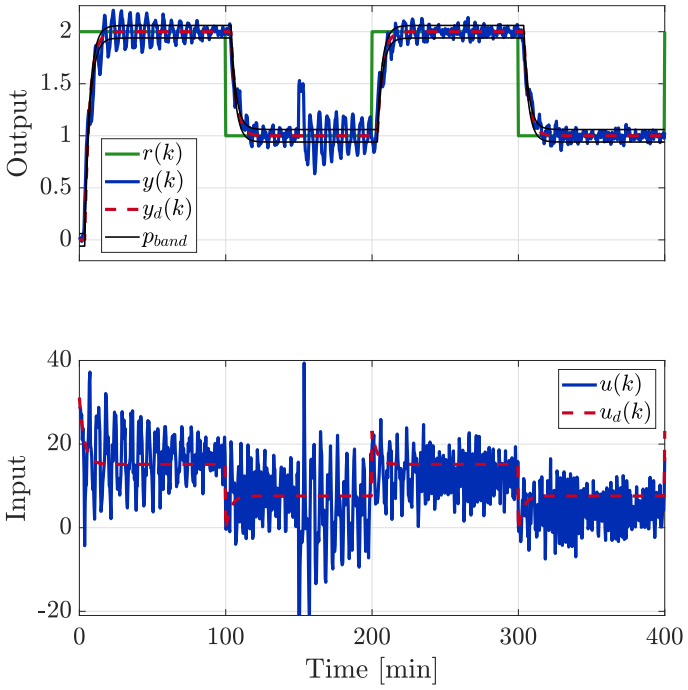


Figure C.1: PI controller for the first-order system.

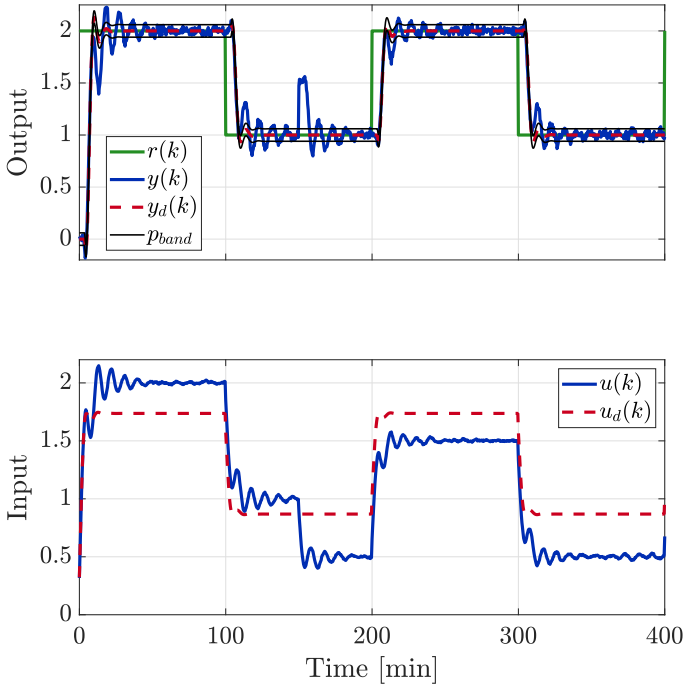


Figure C.2: PI controller for the non-minimum system.

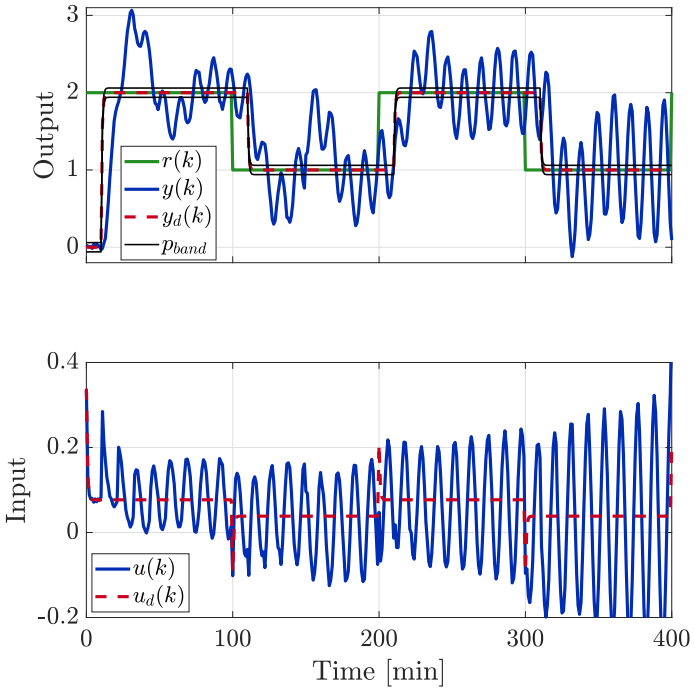


Figure C.3: PI controller for the high-order system.

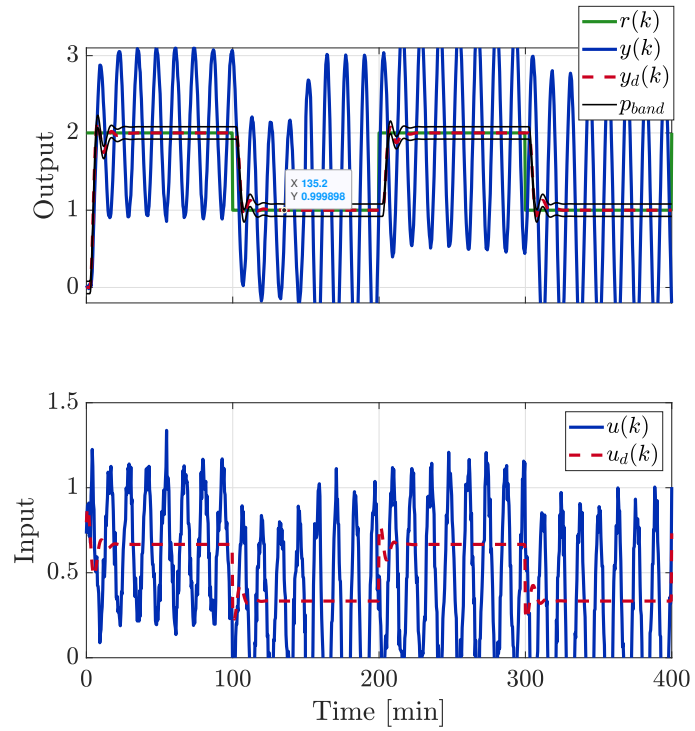


Figure C.4: PI controller for the oscillatory system.

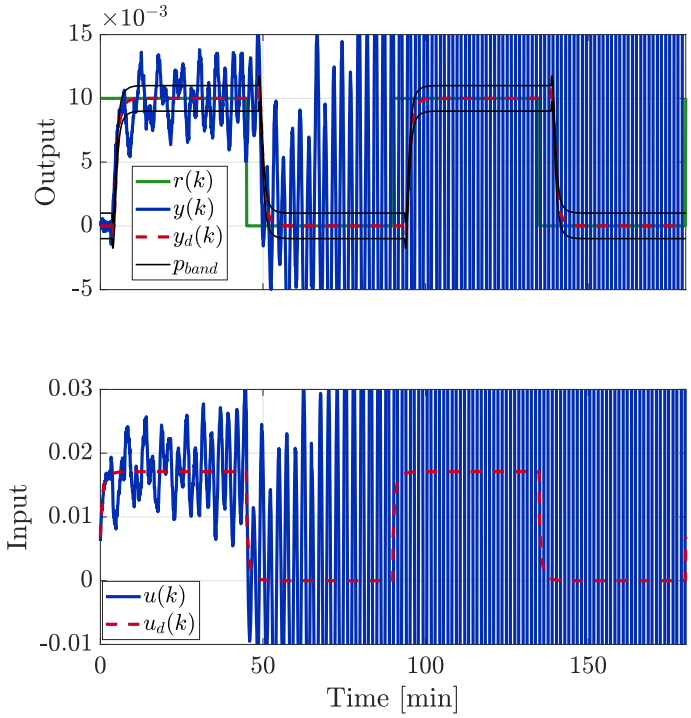


Figure C.5: PI controller for the non-linear system.

(IAE), and the Integral of the Time-weighted Absolute Error (ITAE). These metrics provide different perspectives on the overall control system performance.

Table C.1: Performance metrics for the PI control with the MPM and UD Monitoring Structure.

System	**ISE**	**ITSE**	**IAE**	**ITAE**
First-Order	40.4937	4.5617e+03	83.4652	1.3057e+04
Non-Minimum Phase	27.6742	2.6255e+03	43.5270	5.9989e+03
High-Order	14.8469	1.4733e+03	40.6700	5.2639e+03
Oscillatory	27.1803	1.4070e+03	39.0938	3.1570e+03
Non-Linear	0.0011	0.0623	0.2749	18.1000

Table C.2 shows the performance metrics for the PI control without the Monitoring Structure. These metrics can be compared with those presented in Table C.1 to evaluate the improvement achieved using the Monitoring Structure.

For all systems, there is a noticeable reduction in error metrics (ISE, ITSE, IAE, and ITAE) when the monitoring structure is applied, highlighting the effectiveness of the proposed algorithm in improving control performance.

For instance, in the case of the first-order system, implementing the monitoring algorithm leads to a reduction of about 48% in ISE and 38% in ITSE. Similar trends are observed for the non-minimum phase and high-order systems.

The high-order system's most significant improvement is observed, with a reduc-

Table C.2: Performance metrics for the PI control without the MPM and UD Monitoring Structure.

System	**ISE**	**ITSE**	**IAE**	**ITAE**
First-Order	77.6224	1.1907e+04	133.6909	2.3735e+04
Non-Minimum Phase	29.7460	3.0123e+03	51.0136	7.5649e+03
High-Order	214.1207	6.1301e+04	240.1792	5.9540e+04
Oscillatory	359.5304	8.1906e+04	328.7841	7.1103e+04
Non-Linear	39.8682	6.5075e+03	44.3912	6.6916e+03

tion of approximately 93% in ISE and 98% in ITSE. This substantial improvement illustrates the benefits of the proposed methodology for managing complex high-order systems.

The performance improvements are also noteworthy for the oscillatory and non-linear systems, with reductions in ISE and ITSE of more than 92% and 99%, respectively.

Appendix D

PI Controller Performance on the Temperature Control Lab (TCLab) System

This appendix exhibits the response of the Temperature Control Lab (TCLab) operating solely with a Proportional-Integral (PI) controller, as shown in Figure D.1. This serves as a baseline for comparison with the system's performance when the proposed monitoring algorithm is applied, which is extensively discussed in the main body of the paper.

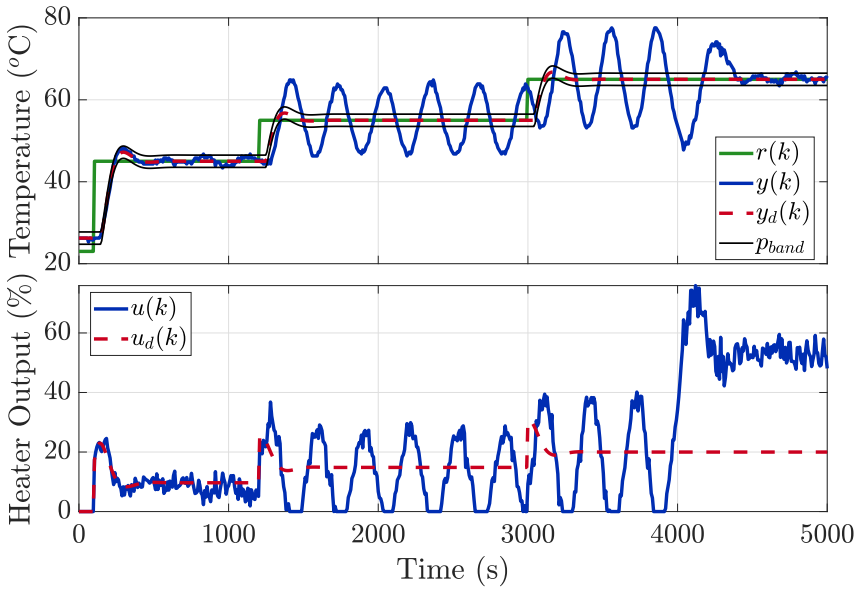


Figure D.1: PI controller for the TCLab.

From Figure D.1, the system performs appropriately at the start, closely tracking

the benchmark signal as there is no MPM. However, as the MPM is gradually introduced into the system's time delay, gain, and time constant, the system begins to oscillate. This oscillatory behavior becomes more prominent when introducing the second and third steps.

At 4000 seconds, a flow of cold air is introduced as a disturbance. Interestingly, in this case, the disturbance helps to counterbalance the effects of the MPM, allowing the FSP to bring the variable back to the setpoint. It is important to clarify that such an occurrence may not always happen. Relying on an external disturbance to improve the system's performance would be impractical and unpredictable, as the improvement observed in this experiment was coincidental.

The difference in the system's performance with and without the monitoring algorithm is evident. The monitoring algorithm allows the system to adapt and respond effectively to changes in system dynamics and disturbances, thereby ensuring the system's stability and performance.

Several integral error metrics were computed for each case to assess the control system's performance with and without the monitoring structure and are presented in Table D.1. The results indicate a noticeable improvement in the system's control performance when the monitoring structure is employed.

Table D.1: Performance Metrics Comparison for the TCLab

Metric	**Without Monitoring**	**With Monitoring**
ISE	1.2977e+05	5.2700e+04
ITSE	3.5486e+08	6.9347e+07
IAE	2.0138e+04	9.0355e+03
ITAE	5.3664e+07	1.5153e+07

Specifically, the ISE shows a reduction of more than 50% when the monitoring structure is applied. This substantial decrease shows the monitoring structure's ability to reduce the accumulated error over the entire control period, reflecting a better tracking performance of the desired setpoints.

The ITSE also witnesses a significant reduction in the monitoring structure. This metric emphasizes the impact of errors that occur later in the process, making it a critical measure in systems where long-term performance is vital. The substantial reduction in ITSE indicates that the control system with monitoring effectively manages to decrease the error over time, providing a more stable long-term response.

The decrease in the IAE and ITAE is also significant when the monitoring structure is used. Both metrics give more weight to significant errors, so the reduction in these metrics implies that the monitoring and self-tuning structure has succeeded in minimizing significant errors, which can be detrimental to the system's stability and performance.

Appendix E

TCLAB Board

The circuit detailed in this appendix shares many features with the original Temperature Control Laboratory (TCLab) design, including two sensors and two actuators. Additionally, it incorporates a pair of current sensors and a temperature sensor, increasing the total to five sensors and two actuators. These extensions enhance the functionality of the system and enable a wider range of didactic activities.

The system is a control scheme that uses a microcontroller mounted on an Arduino Uno platform. This microcontroller connects to a personal computer on one side and a thermal system on the other. The thermal system comprises two actuators, two direct temperature sensors on the actuators, and an ambient temperature sensor. Moreover, two current sensors measure the current consumed by the actuators.

This setup allows for practices in modeling, identification, and multivariable control. It provides students with a portable system that can interact with software written in Matlab, Simulink, Python, or C (Arduino). The circuit is shown in Figure E.1.

Further information about the board can be found in the GitHub repository: https://github.com/sergioacg/TCLAB_CAE. This repository contains Gerber files for producing the circuit board. Gerber files are standard files used in PCB manufacturing to provide comprehensive layout information, including copper layers, solder mask, legend, and drill and route data.

The repository also contains Matlab scripts and Simulink diagrams to carry out initial control practices using the TCLAB board. This collection of resources is designed to aid users in effectively utilizing the TCLAB for various control engineering tasks. Also included is the Arduino code, which reads the sensor data and provides for temperature and current levels. It controls the transistors by PWM signals. And it allows the board to communicate with the computer. This way, the board can work with other software like Matlab, Simulink, or Python. This Arduino code is a key part of how the TCLAB board works.

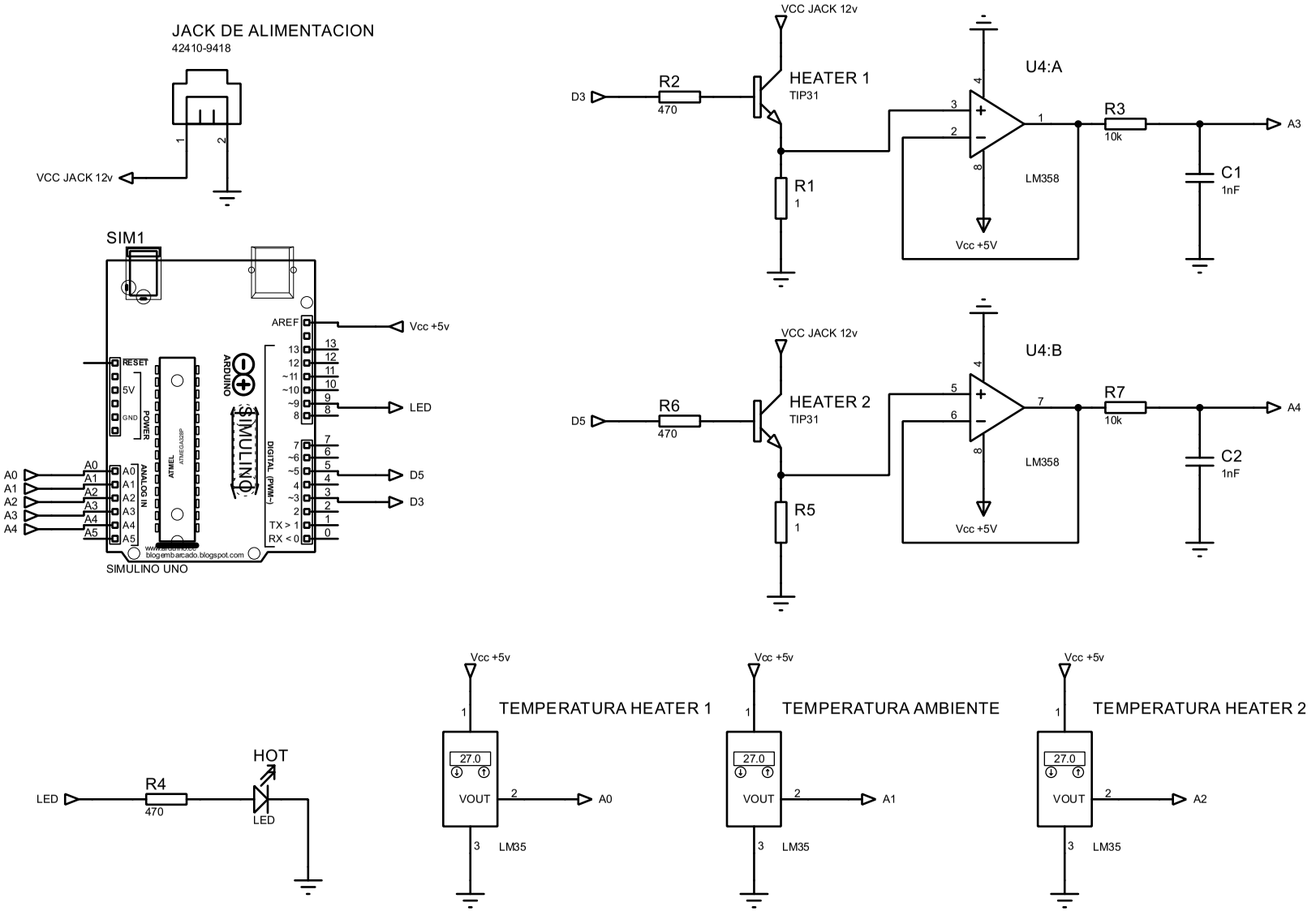


Figure E.1: TCLab circuit.