UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Lucian Ribeiro da Silva

Control of industrial processes: contributions for analysis and implementation of controllers for processes with dead time, constraints, and modeling errors

Florianópolis
2023

Lucian Ribeiro da Silva

Control of industrial processes: contributions for analysis and implementation of controllers for processes with dead time, constraints, and modeling errors

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do título de doutor em Engenharia de Automação e Sistemas.
Orientador: Prof. Rodolfo César Costa Flesch, Dr.
Coorientador: Prof. Julio Elias Normey-Rico, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Lucian Ribeiro da
Control of industrial processes : contributions for analysis and implementation of controllers for processes with dead time, constraints, and modeling errors / Lucian Ribeiro da Silva ; orientador, Rodolfo César Costa Flesch, coorientador, Julio Elias Normey-Rico, 2023.
109 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. PID. 3. Controle preditivo baseado em modelo. 4. Atraso de transporte. 5. Restrições. I. Flesch, Rodolfo César Costa. II. Normey-Rico, Julio Elias. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Lucian Ribeiro da Silva

Control of industrial processes: contributions for analysis and implementation of controllers for processes with dead time, constraints, and modeling errors

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Lucíola Campestrini, Dr.
Universidade Federal do Rio Grande do Sul

Prof. Daniel Martins Lima, Dr.
Universidade Federal de Santa Catarina

Prof. Darci Odloak, Dr.
Universidade de São Paulo

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em Engenharia de Automação e Sistemas.

Documento assinado digitalmente
Julio Elias Normey Rico
Data: 30/08/2023 18:00:05-0300
CPF: ***.840.859-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Julio Elias Normey-Rico, Dr.
Coordenador do Programa

Documento assinado digitalmente
Rodolfo Cesar Costa Flesch
Data: 30/08/2023 15:55:38-0300
CPF: ***.466.879-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Rodolfo César Costa Flesch, Dr.
Orientador

Florianópolis, 2023.

ACKNOWLEDGEMENTS

To my family, for their encouragement and understanding during my academic journey.

To my fiancée Bruna, for being my unconditional partner, friend, and confidant. Her love, support, and patience were essential for me to complete this work. Her kind and encouraging words kept me motivated and focused on my academic goals.

To my dear dog Coquinha, for her constant presence during this last year. Her loyalty, joy, and love were a source of comfort and happiness for me. Her company helped me alleviate stress and anxiety during the difficult moments of my academic journey.

To my advisor, Professor Rodolfo, and my co-advisor, Professor Julio, for all the support, guidance, patience, and valuable teaching throughout the elaboration of this thesis. Without their help, it would not be possible to achieve this result. You are inspiring examples of teachers and researchers committed to academic excellence and the formation of new researchers.

To my friends and colleagues from the doctoral program for their companionship and friendship.

To CAPES and CNPq for their financial support.

"The essence of mathematics is not to make simple things complicated, but to make complicated things simple."
(Stan Gudder)

RESUMO

Os processos encontrados em ambiente industrial tipicamente apresentam características que aumentam a complexidade do projeto do sistema de controle. Essas características podem incluir atraso de transporte, dinâmicas complexas, ruído de medição, não linearidades, incertezas de modelagem, dentre outras. No contexto acadêmico da área de controle, o projeto de sistemas de controle para processos com tais características é um tema de pesquisa em constante evolução. Com o objetivo de contribuir para solucionar alguns dos problemas encontrados na indústria, este trabalho apresenta uma abordagem abrangente composta por análise comparativa entre diferentes estratégias de controle, ferramenta de simulação, e uma proposta de estrutura de controle capaz de fornecer desempenho ótimo. A análise realizada neste trabalho consiste em comparar e avaliar os tipos de controladores mais comumente utilizados na indústria, fornecendo diretrizes para que o engenheiro possa selecionar a melhor estrutura de controle com base nas características específicas do processo. A ferramenta proposta possui uma interface amigável e é capaz de comparar e validar, de maneira intuitiva e simples, o desempenho e a robustez de várias estruturas de controle usadas em aplicações industriais. Por fim, a proposta de controle apresentada consiste em uma sintonia de controle PID ótimo, para processos cuja dinâmica possa ser aproximada por modelos de primeira ou segunda ordem, seja ela estável, integradora ou instável, em conjunto com uma estrutura que permite o tratamento de restrições de entrada. A abordagem proposta também pode ser utilizada para a sintonia do controlador primário de estratégias de compensação de atraso. Essa proposta é capaz de apresentar desempenho equivalente ao de um controlador preditivo baseado em modelo, mas sem a necessidade de um otimizador on-line, o que torna a solução proposta conveniente para ser utilizada em sistemas com hardware de baixo custo.

Palavras-chave: PID. Controle preditivo baseado em modelo. Atraso de transporte. Processos industriais.

RESUMO EXPANDIDO

Introdução

A maioria dos processos encontrados na indústria apresentam quatro características que aumentam a complexidade da estrutura de controle a ser utilizada: atraso de transporte, restrições, ruído de medição e incertezas de modelagem. O atraso de transporte é definido como o tempo necessário para que a ação de controle surta algum efeito na saída do processo. A presença de atraso em uma malha de controle diminui a margem de fase do sistema, o que torna a sintonia do controlador mais complicada. As restrições são limitações, geralmente físicas ou de projeto, presentes na maioria dos processos, como, por exemplo, os limites de abertura de uma válvula. Essas limitações podem inserir não linearidades no sistema, degradando o desempenho em malha fechada do controlador se ele não considerá-las em seus cálculos. O ruído de medição está relacionado a variações não previsíveis ou flutuações na medição da saída de um processo. Ele pode se originar a partir de várias fontes, como, por exemplo, interferências elétricas nos sensores ou atuadores, e pode interferir na estabilidade do sistema, dependendo da região de operação.

Para lidar com os problemas causados pelo atraso, tipicamente são utilizados controladores proporcional-integral-derivativo (PID), compensadores de atraso de transporte (DTCs, do inglês *dead-time compensators*) ou controladores preditivos baseados em modelo (MPCs, do inglês *model predictive controls*). Já no contexto de restrições, as abordagens mais utilizadas para mitigar os efeitos causados envolvem o uso de estruturas chamadas de anti-*windup* ou abordagens conhecidas como *governors*. Além disso, abordagens MPC também podem tratar tanto restrições quanto atraso de transporte em sua formulação original. Entretanto, ao considerar restrições, as abordagens MPC demandam um processo de otimização em cada instante de amostragem, limitando seu uso a certos tipos de plantas, que tipicamente apresentam dinâmicas que permitem amostragem na ordem de segundos, minutos ou horas.

Dentre os desafios centrais no projeto de sistemas de controle para processos com características comumente encontradas na indústria, também se destacam duas questões: como escolher a estrutura mais adequada para controlar um processo com essas características? E como aprimorar a capacitação de engenheiros de controle para lidar com estruturas complexas, como DTC e MPC? Diante deste cenário, neste trabalho foram delineadas três linhas de pesquisa com o objetivo de preencher algumas lacunas desse contexto:

- um estudo comparativo entre PID, DTC e MPC ao controlar processos monovariáveis com atraso de transporte, considerando aspectos importantes como desempenho, robustez e tratamento de restrições, visando fornecer diretrizes para facilitar a escolha da estratégia mais conveniente com base nas características do processo;

- o desenvolvimento de uma ferramenta interativa com interface gráfica e de fácil uso para o projeto, simulação e análise de sistemas em malha fechada, considerando controladores PID, DTC e MPC para processos com características industriais;

- a formulação de uma sintonia PID com uma abordagem semelhante àquela empregada nas técnicas *governor*, baseada em um controlador MPC considerando horizontes de controle curtos, capaz de fornecer desempenho ótimo considerando processos com restrições de entrada, sem a necessidade de um procedimento de otimização on-line.

Objetivos

O objetivo geral deste trabalho é investigar abordagens de sistemas de controle aplicadas a sistemas industriais com características como atraso, restrições, erros de modelagem e ruído de medição, e desenvolver soluções práticas, ferramentas e *insights* com base em desafios comuns e estruturas de controle típicas usadas nesse contexto.

As principais contribuições desta pesquisa visam facilitar a análise e seleção da estrutura de controle mais adequada com base nas características do processo, fornecendo orientações valiosas para o projeto e implementação de sistemas de controle em aplicações industriais.

Metodologia

Inicialmente, foi realizada uma revisão abrangente do estado da arte das estruturas de controle empregadas em processos com atraso e restrições. Com base nesse panorama, foi conduzida uma análise comparativa das três principais abordagens utilizadas para o controle de processos nesse contexto: estruturas PID, DTC e MPC. O objetivo central foi de fornecer informações que orientassem a seleção da estrutura de controle mais apropriada de acordo com as características intrínsecas de cada processo. Essa análise englobou uma variedade de situações, incluindo processos estáveis, integradores e instáveis. Primeiramente, consideraram-se apenas processos com atraso de transporte; posteriormente, foram incorporadas restrições. As considerações também abrangeram fatores como erros de modelagem e ruído de medição.

A segunda etapa deste trabalho teve como foco a implementação de uma ferramenta interativa capaz de sintonizar, analisar e simular estruturas de controle aplicadas a processos monovariáveis considerando diferentes cenários. Essa ferramenta foi desenvolvida no ambiente de software MATLAB®.

A terceira e última parte do trabalho teve como objetivo principal propor uma técnica de sintonia de controladores PID fundamentada com base em um método MPC, aliada a uma estrutura análoga às abordagens *governor*. A ideia é que a proposta possa ter um desempenho equivalente ao do MPC sem a necessidade de um otimizador on-line.

Resultados e Discussão

A análise comparativa entre PID, DTC e MPC foi conduzida com base em quatro estudos de caso: um processo estável com constante de tempo dominante, um processo estável com atraso dominante, um processo integrador e um processo instável. Com base nesses cenários, foram comparadas as abordagens PID e preditor de Smith filtrado. Nos três primeiros cenários, ficou evidente que, quando um sistema robusto é necessário, devido, por exemplo, a uma modelagem que não captura adequadamente a dinâmica do sistema, o PID se mostra a escolha mais apropriada. Nesses casos, o PID oferece um desempenho praticamente equivalente ao do preditor de Smith filtrado, além de ter uma implementação prática mais simples. Entretanto, quando o modelo da planta é capaz de representar de forma precisa a dinâmica do processo e uma sintonia voltada para o desempenho é viável, o preditor de Smith filtrado se mostra mais adequado. Para o caso do processo instável, o preditor de Smith filtrado demonstrou superioridade tanto em desempenho quanto em robustez, seja para sintonias agressivas ou robustas. Considerando o caso com restrições, foi realizado um estudo de caso experimental comparando o desempenho de um PID com anti-*windup* e uma abordagem MPC. Os resultados mostraram que o PID com anti-*windup* obteve um resultado praticamente equivalente ao de um MPC, no entanto com uma simplicidade de implementação muito maior.

Em relação à ferramenta desenvolvida, suas características englobam: sintonia simplificada de controladores amplamente adotados na indústria; análise do desempenho da resposta em malha fechada de processos, incorporando elementos comuns em aplicações práticas; possibilidade

de incorporação de ação anti-*windup* na estrutura dos controladores; avaliação de robustez, abrangendo índices clássicos utilizados na indústria. Essas características tornam a ferramenta proposta uma opção valiosa para ensinar conceitos essenciais da engenharia de controle.Um estudo de caso envolvendo um processo integrador com atraso e restrições foi apresentado para ilustrar de forma mais clara as funcionalidades da ferramenta.
Por fim, a abordagem pioneira de sintonia de PID, aliada a uma estrutura adicional capaz de tratar restrições de entrada, demonstrou a capacidade de atingir resultados comparáveis aos de uma abordagem MPC, considerando situações com restrições. Isso foi alcançado com um custo computacional significativamente menor. A proposta foi capaz de obter a solução ótima cerca de 600 vezes mais rápido que um MPC, considerando o tempo médio de cálculo da ação de controle, e aproximadamente 370 vezes mais rápido considerando o pior cenário de tempo de execução.

Considerações Finais

A partir do estudo comparativo entre estruturas PID, DTC e MPC foi possível concluir que a escolha da melhor estratégia de controle está mais relacionada à qualidade do modelo obtido do processo (em relação à representação da dinâmica da planta) do que à magnitude do atraso. Em contextos industriais que demandam soluções robustas, o controle PID com uma estrutura de ação anti-*windup* é capaz de oferecer um desempenho tão bom quanto, ou até mesmo superior a, estratégias mais complexas, como DTC e MPC. No entanto, em situações em que a robustez não é uma preocupação central e um desempenho rápido é possível e desejável, optar por uma abordagem DTC ou MPC se torna mais vantajoso, mesmo em cenários com valores pequenos de atraso.
Com base nas funcionalidades implementadas na ferramenta desenvolvida na segunda parte deste trabalho, é possível afirmar que ela se destaca como um recurso de grande valor tanto para estudantes quanto para engenheiros de controle. Além de facilitar a compreensão de conceitos complexos relacionados às estruturas de controle aplicadas a sistemas com características industriais, a ferramenta também oferece suporte na seleção do método mais adequado a ser implementado com base na particularidade do processo.
A sintonia inédita de controladores PID com base em estruturas MPC demonstrou ser capaz de calcular com sucesso a mesma solução ótima do método MPC, no entanto, sem a necessidade de um otimizador on-line. Essa característica torna essa abordagem especialmente adequada para sistemas com dinâmicas rápidas ou com recursos de hardware limitados, como, por exemplo, microcontroladores de baixo custo.
Portanto, os resultados desta tese estão alinhados com os objetivos estabelecidos e proporcionam contribuições significativas para a análise e o projeto de controladores destinados a sistemas com características frequentemente encontradas na indústria, como atraso de transporte, restrições nas variáveis controladas e manipuladas, erros de modelagem e ruído de medição. Os desafios relacionados ao controle de tais processos são abordados, e diretrizes e ferramentas são apresentadas de maneira a facilitar a seleção e a implementação das estruturas de controladores mais apropriadas com base nas particularidades do processo.

Palavras-chave: PID. Controle preditivo baseado em modelo. Atraso de transporte. Processos industriais.

ABSTRACT

Processes found in industry typically exhibit characteristics that increase the complexity of the control system design. These characteristics can include dead time, complex dynamics, measurement noise, non-linearities, modeling uncertainties, among others. In the academic context of control engineering, the design of control systems for processes with such characteristics is a constantly evolving research topic. With the aim of contributing to solving some of the problems found in industry, this work presents a comprehensive approach composed of a comparative analysis of different control strategies, a simulation tool, and a proposal for a control structure capable of achieving optimal performance. The analysis conducted in this work involves comparing and evaluating the most commonly used control strategies in industry, providing guidelines for the engineer to select the best control structure based on the specific characteristics of the process. The proposed tool has a user-friendly interface and is capable of intuitively and simply comparing and validating the performance and robustness of various control structures used in industrial applications. Finally, the proposed control approach consists of an optimal PID control tuning for processes whose dynamics can be approximated by first or second-order models, whether they are stable, integrating, or unstable, along with a structure that allows for handling input constraints. The proposed approach can also be used for tuning the primary controller of dead-time compensation strategies. This proposed method is capable of delivering performance equivalent to a model predictive controller, but without the need for an online optimizer, which makes the proposed solution convenient to be used in systems with low-cost hardware.

Keywords: PID. Model predictive control. Dead time. Industrial processes.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS

$A(z^{-1})$	polynomial associated with the poles of the plant
$B(z^{-1})$	polynomial associated with the zeros of the plant
$C(s)$	controller
$C_{\mathrm{FSP}}(z)$	discrete-time filtered Smith predictor primary controller
$C_{\mathrm{GPC}}(z)$	discrete-time GPC equivalent controller
$C_{\mathrm{PID}}(z)$	discrete-time PID controller
$C_{\mathrm{PI}}(z)$	discrete-time PI controller
$C_{\mathrm{SP}}(s)$	Smith predictor primary controller
$C_{\mathrm{SP}}(z)$	discrete-time Smith predictor primary controller
$C(z)$	discrete-time controller
d	discrete dead time
d_n	discrete nominal dead time
D_M	delay margin
$e(k)$	discrete-time error signal
$e_{cf}(k)$	correction factor used in back-calculation method
$e_p(t)$	prediction error
$F(s)$	reference filter
$F(z)$	discrete-time reference filter
$F_{\mathrm{GPC}}(z)$	discrete-time reference filter of the equivalent structure of GPC
$F_{\mathrm{PID}}(z)$	discrete-time reference filter of a 2DOF PID structure
$F_r(s)$	robustness filter
$F_r(z)$	discrete-time robustness filter
$G(s)$	dynamics of the process
$G(z)$	discrete-time dynamics of the process
$G_n(s)$	fast process model
$G_n(z)$	discrete-time fast process model
$H_{yr}(s)$	reference response transfer function
$H_{yq}(s)$	disturbance response transfer function
J	cost function of the integral of absolute error
J_{MPC}	MPC objective function
K	gain of the primary controller of filtered Smith predictor
L	dead time
L_n	nominal dead time
M_s	maximum sensitivity
N	prediction horizon
N_1	start of prediction horizon
N_2	end of prediction horizon

N_u	control horizon
$P(s)$	real plant
$P_i(s)$	integrating process model
$P_n(s)$	nominal process model
$P_n(z)$	discrete-time nominal process model
$P_s(s)$	stable process model
$P_u(s)$	unstable process model
$P(z)$	discrete-time process model
$P_m(s)$	first-order plus dead time model
$q(t)$	load disturbance
$R(s)$	Laplace transform of the reference signal
R_{UC}	feasible region of a polytope
s	Laplace variable
T_0	closed-loop time constant
T_d	derivative time constant of PID controller
T_i	integral time constant of PID controller
T_s	sampling period
T_t	back-calculation tracking-time parameter
t	time
$u(k)$	discrete-time control action
$u_i^*(k)$	recalculated integral term of the controller
$u_r(k)$	discrete-time control action applied to the plant
$U(z)$	discrete-time control input
$U_{\max}$	maximum control action to maintain the magnitude and increment of control action and the process output prediction in feasible region
$U_{\min}$	minimum control action to maintain the magnitude and increment of control action and the process output prediction in feasible region
$u_{\max}$	maximum value of the control action
$u_{\min}$	minimum value of the control action
$u_{\mathrm{y}_{\max}}$	maximum value of the control action to maintain the process output prediction in feasible region
$u_{\mathrm{y}_{\min}}$	minimum value of the control action to maintain the process output prediction in feasible region
$y(k)$	discrete-time process
$y(k+j\|k)$	discrete-time predicted process output at time $k+j$
$y_p(t)$	prediction of the process output
$y_{\max}$	maximum limit of the process output
$y_{\min}$	minimum limit of the process output
$Y(s)$	Laplace transform of the process output

$y(t)$	process output
$Y(z)$	z-transform of the process output
z	z-transform variable
α	filter parameter of the derivative action
β	parameter to eliminate the zero of the FSP equivalent controller
δ	reference tracking error weight
γ	slope of affine function
ψ	intercept
τ	process time constant
$\mathbf{1}$	vector of ones in $\mathbb{R}^{N_u}$
$\tilde{\mathbf{1}}$	vector of ones in $\mathbb{R}^{N}$
$\mathbf{A_c}$	matrix of constraints
$\mathbf{b_c}$	vector of constraint bounds
$\tilde{\mathbf{A}}_\mathbf{c}$	matrix of constraints in the transformed coordinates
$\tilde{\mathbf{B}}_\mathbf{c}$	vector of constraint bounds in the transformed coordinates
$\mathscr{Z}$	z-transform
$\mathbf{V}$	term which is function of the unconstrained optimal control actions
$\Delta\mathbf{u}^\star_{\mathrm{UC}}$	vector of unconstrained optimal control actions
$\Delta\mathbf{u}^\star$	vector of constrained optimal control actions
$\Delta\tilde{\mathbf{u}}^\star_{\mathrm{UC}}$	vector of unconstrained optimal control actions in the transformed coordinates
$\Delta\tilde{\mathbf{u}}^\star$	vector of constrained optimal control actions in the transformed coordinates
$\tilde{R}_{\mathrm{UC}}$	feasible region of a polytope in the transformed coordinates

LIST OF ABBREVIATIONS AND ACRONYMS

2DOF	two-degree-of-freedom
AW	anti-windup
BC	back-calculation
CARIMA	controlled auto-regressive integrated moving average
CSPS	constrained SISO-process simulator
CSG	control signal governor
DTC	dead-time compensator
DTC-GPC	dead-time compensator generalized predictive control
DMC	dynamic matrix control
D_M	delay margin
EG	error governor
ER	error recalculation
EHAC	extended horizon adaptive controller
FOPDT	first-order plus dead time
FSP	filtered Smith predictor
GPC	generalized predictive control
GPCIT	generalized predictive control interactive tool
GUIDE	graphic user interface design
IPDT	integrating plus dead time
IFAC	international federation of automatic control
IAE	integral of absolute error
MIMO	multi-input multi-output
MPC	model predictive control
M_s	maximum sensitivity
PI	proportional-integrative

PID	proportional-integrative-derivative
P_M	phase margin
QDMC	quadratic dynamic matrix control
QP	quadratic programming
RG	reference governor
RHC	receding horizon control
R_I	robustness index
SBAI	simpósio brasileiro de automação inteligente
SDD	stable delay dominant
SMOC	shell multivariable optimization control
SLD	stable lag dominant
SISO	single-input single-output
SP	Smith predictor
TRIAC	triode for alternating current
TT	temperature transducer
TC	temperature controller
UPC	unified predictive control
UFOPDT	unstable first-order plus dead time
ZOH	zero-order hold
IA	incremental algorithm

CONTENTS

1 INTRODUCTION

Several industrial processes present three main characteristics that increase the complexity of the control system design: transport delay (or dead time), constraints, and measurement noise. In general, dead time can be defined as the time required for a change in the control signal to have an effect on the process output (CAMACHO; BORDONS, 2013). Examples of processes with dead time can be observed in several domains, and some of them are distillation columns, heat exchangers, photovoltaic solar panels, among others (NORMEY-RICO; CAMACHO, 2007). Constraints are limitations, usually physical or imposed by design, that the process presents, such as the limits of a valve opening. These limitations not only introduce nonlinearities to the system, making it difficult to control in closed loop, but can also cause damage to the system (ÅSTRÖM; WITTENMARK, 2013). Measurement noise refers to unwanted and unpredictable variations or fluctuations in the measured process output (ELLIS, 2012). It can originate from various sources, such as measurement errors, electrical interference in the sensors or transducers, or system nonlinearities, and can have significant implications for the performance and stability of the control system, depending on its operating range (KUO; MORGAN, 1996).

In the literature, various proposals and control tuning techniques are presented to address the problem generated by dead time. Among them, three approaches are widely disseminated and used in industry: proportional-integrative (PI) or proportional-integrative-derivative (PID) controllers, which are very popular controllers found in approximately 90% of the applications in industry (OVIEDO; BOELEN; OVERSCHEE, 2006; CASTRO et al., 2016); dead-time compensator (DTC), which makes use of a predictor structure to reduce the degradation of the system performance caused by the presence of dead time (NORMEY-RICO; CAMACHO, 2007); and model predictive control (MPC), which is a method that uses the information of future output predictions in order compute a control action based on an optimization procedure (CAMACHO; BORDONS, 2013). However, when it comes to constraints, only MPC is capable of explicitly addressing them in its original formulation.

The presence of dead time in processes dynamics causes a reduction in the phase margin of the system, thus increasing the complexity of the tuning (VISIOLI; ZHONG, 2010). In most instances, models such as the first-order plus dead time (FOPDT), integrating plus dead time (IPDT), and unstable first-order plus dead time (UFOPDT) have proven to be effective in representing the dynamic behavior of single-input single-output (SISO) systems with delay. Over the years, various PID tuning rules have been developed for processes of these types in order to attain a desirable equilibrium in the trade off between robustness and performance, while maintaining a relatively straightforward structure (PANDA; YU; HUANG, 2004; NORMEY-RICO; GUZMÁN, 2013). This is one of the primary reasons why PID controllers have gained widespread adoption in industry (ÅSTRÖM; HÄGGLUND, 2001; SKOGESTAD, 2018).

A control method that became quite popular and is still widely used in industry to compensate for some of the effects caused by dead time in the control loop was proposed by

Smith (1957). This strategy, known as the Smith predictor (SP), was originally developed for SISO processes and explicitly uses the process model without the dead time (known as fast model) to make a prediction of the output observed in the process after the dead time. This procedure has the capability to eliminate the undesirable effects of dead time in the closed-loop characteristic equation in the case of a perfect modeling of the process, thus addressing the problem of phase margin reduction (SMITH, 1957). This type of approach became known in the literature as a dead-time compensator (DTC) and is capable of improving the closed-loop response compared to classical control techniques (NORMEY-RICO; CAMACHO, 2007). Many extensions of the original SP are presented in the literature to broaden its ideas to a wider class of problems, such as open-loop unstable processes or multi-input multi-output (MIMO) processes (KAYA, 2003; SANZ; GARCÍA; ALBERTOS, 2018; LIU et al., 2018).

According to Normey-Rico and Camacho (2007), in practice, all processes are subject to physical limitations. One of these limitations, which is quite common in industrial processes, is known as actuator saturation, which, depending on the process operating point, can introduce nonlinearities into the system caused by the discrepancy between the signal calculated by the controller and the signal applied to the plant. This characteristic can degrade the performance of the controller if it has slow or unstable modes, causing large oscillations in the process output, high settling time, and even system instability (DOYLE; SMITH; ENNS, 1987; HIPPE, 2006). Saturation constraints play a very important role in process control, both from safety and economic perspectives. In terms of safety, there are reports of very serious accidents in which actuator saturation was the main cause, such as the accident of the JAS 39 Gripen aircraft model in 1993 (RUNDQWIST; STÅHL-GUNNARSSON; ENHAGEN, 1997). Another well-known example was the Chernobyl nuclear disaster in 1986, in which the saturation of the actuators at the facility was one of the reasons that caused the melting of one of the units at the plant (STEIN, 2003). From an economic perspective, in order to avoid the problems generated by actuator saturation, in most cases, companies design their control systems using oversized actuators, which end up increasing the cost of the control system project (GALEANI et al., 2009). The degradation of the controller performance caused by saturation became known in literature as windup.

One of the most popular approaches to deal with the windup became known as anti-windup (AW) techniques. Basically, AW strategies are add-on structures that can be included into a control system that was designed ignoring the saturation limits of the actuator (O'DWYER, 2009). Thus, it becomes easy to apply this type of approach in systems where the implemented controller has good performance, except when it operates in the saturation region, and without the necessity to retune the controller. As almost all processes in industry present actuator limitations, many AW strategies were proposed in the literature. For a review of the main contributions in this subject, the reader may refer to the survey works by Kothare et al. (1994), Zaccarian and Teel (2011), Tarbouriech and Turner (2009).

Other constraint handling methods that are widely discussed in the literature are the governor approaches, with emphasis placed on the reference governor (RG) and error governor

(EG) approaches (KAPASOURIS; ATHANS; STEIN, 1988; GILBERT; TAN, 1991). Both RG and EG are add-on schemes that are capable of dealing with input constraints by modifying the reference signal (for the RG approach) or the error signal (for the EG approach) of the pre-stabilized system to a safe value that guarantees no violation of the constraints. As the AW methods, governors approaches can also be added into systems with well-tuned controllers in a separated structure, with the aim of reducing the effects of actuator saturation.

Another approach that is capable of dealing with process constraints is the MPC technique. MPC is a strategy that explicitly uses the plant model to compute the control signal that minimizes an objective function, in which constraints can be considered, with the aim of optimizing the future behavior of the process output and also the control effort over a prediction horizon (WANG, 2009). A major advantage of MPC is that, in addition to compensating the implicit dead time, it can deal with several types of process constraints, such as constraints on the amplitude and rate of variation of the control signal and also on the process output. The optimization procedure used to compute an optimal control action takes these constraints into account, which turns MPC a popular approach for achieving optimal process operation and economic benefits (RODRIGUES; ODLOAK, 2000).

Applications in the petrochemical, automotive, and robotics industries are examples in which predictive controllers are used successfully (RODRIGUES; ODLOAK, 2000). A well-known drawback of MPC, in its original formulation, is that an online optimizer is necessary to compute the control action at each sampling instant, which may require a considerable computational effort depending on the process characteristics, thus limiting its use to high-cost computational systems or to processes with slow dynamics (SILVA; FLESCH; NORMEY-RICO, 2020).

Although governor approaches and MPC strategies provide good performance when used to control processes with constraints, there must be a high cost investment in resources and expertise to implement and supervise these advanced control methods (FORBES et al., 2015). On the other hand, in most practical applications, the well-known PID with an AW method is capable of providing good results in terms of performance and robustness, and is also easier to implement than more advanced techniques (SILVA; FLESCH; NORMEY-RICO, 2020). The literature presents a variety of methods to design PID controllers based on MPC methods (TAN; HUANG; LEE, 2000; TAN et al., 2002; TAKAO; YAMAMOTO; HINAMOTO, 2004; SATO, 2010). However, the performance of these controllers was only compared in processes that do not consider any type of constraint. On the other hand, the works of Miller et al. (1999) and Uduehi, Ordys and Grimble (2001) consider input constraints, but the performance of the PID controller is not optimal as the one provided by the MPC method.

Besides the numerous challenges faced in designing controllers for industrial processes, two particularly significant issues must also be addressed: how to choose the most convenient strategy based on the process characteristics, and how to enhance the qualifications of the control engineers workforce.

In the past, when only simple and low-memory hardware were available at afford-

able prices, more complex control strategies, such as DTC and MPC, were only considered for complex plants, such as MIMO, unstable and non-minimum phase processes, where the need for high performance justified higher implementation costs. Nowadays, the availability of low-cost hardware allows the implementation of advanced controllers even for processes with simpler dynamics. Recently, several investigations have been carried out to study the application of MPC in SISO processes, for example, in smart grid systems (HALVGAARD et al., 2012; MBUNGU et al., 2016), heating processes (VALENCIA-PALOMO; ROSSITER, 2011; ZHANG et al., 2017), pneumatic processes (OSMAN et al., 2012; BAGYAVEERESWARAN et al., 2016), and power converters and drives (VAZQUEZ et al., 2017; IPOUM-NGOME et al., 2019; CAVANINI; CIMINI; IPPOLITI, 2019). Therefore, it is interesting to analyze the advantages of using these more advanced control structures over the classical PID solutions, considering the real situation in industry.

Comparisons between PID and DTC have been presented in the literature since the 1980s. Rivera, Morari and Skogestad (1986) present a comparison between a PID designed for FOPDT models and an ideal DTC (i.e. with infinity gain) and show that for large ratios of dead time and time constant, it is possible to obtain significant performance improvement by using a DTC instead of a PID. In Normey-Rico and Camacho (2007), a discussion on when to use a DTC or PID is presented, showing that the performance gains of DTC are more affected by the error of the estimate of dead time than by the absolute value of the dead time. In Sha'aban, Lennox and Laurí (2013), a comparison between PID and MPC strategies considering model mismatch and measurements noise is presented, showing that for first order processes the PID performance degrades almost linearly when dead time exceeds twice of the time constant, and for second order processes the same behavior is observed when dead time exceeds about 10% of the equivalent time constant. For these cases, MPC presented significantly better performance than PID. In Salem and Mosaad (2015), a comparison between PID and MPC is presented considering no modeling errors. The results show that for all the case studies MPC presented better performance than PID. In Sudibyo et al. (2012), the performances of a PI and an MPC are compared for a reactive distillation MIMO process, showing that the MPC provides better results than the PI controller. A comparative analysis between a MIMO filtered Smith predictor (FSP) and a PI controller is presented in Flesch, Santos and Normey-Rico (2012), showing that the main advantage of FSP over the PI strategy is related to the disturbance rejection response of the system if the modeling error is small. While the use of DTC, MPC, and other advanced control strategies can lead to better performance in certain cases, the cited works highlight the importance of PID controllers due to their broad applicability across a wide range of processes. Nonetheless, careful consideration should be given to the error in estimating dead time and other model uncertainties, as well as the impact of noise measurements and constraints, when selecting a control strategy. Thus, theoretical analysis and simulation studies must be conducted to develop the most suitable solution for a given problem.

Advanced mathematics and control engineering concepts used to solve many of these practical problems presented by systems with industrial characteristics are generally involved.

Many students and engineers in the field have difficulties to assimilate the mathematical concepts behind each control strategy, which leads to a decrease in student motivation (MÉNDEZ et al., 2006). In such cases, user friendly tools, which enable the students to simulate systems with different control structures and strategies considering characteristics commonly found in practical applications, can be useful in teaching and training or as supporting material. In the last decade, several interactive tools were proposed to facilitate the teaching of concepts in control engineering. In the work of Guzman, Berenguel and Dormido (2005), a generalized predictive control interactive tool (GPCIT) was proposed to help students to understand basic and advanced concepts of generalized predictive control (GPC) strategy. An equivalent tool for teaching PID concepts was proposed in (GUZMÁN et al., 2006). Some tools also support the comparison between different control structures, such as the web-based tool for analysis and simulation of automatic control systems using PID or state feedback controllers presented in (MÉNDEZ et al., 2006), or the interactive tools to facilitate the design and analysis of PID, DTC, and MPC controllers for processes with dead time proposed in (da COSTA FILHO; NORMEY-RICO, 2009; NORMEY-RICO et al., 2009).

Thus, based on the previous analysis about the challenges of control of processes with industrial characteristics, in this work, three lines of research were made aiming to fill some gaps of this context:

1. a comparative study between PID, DTC, and MPC when controlling SISO dead-time processes, considering important aspects, such as performance, robustness and handling of constraints, aiming to provide guidelines to facilitate the choice of the most convenient strategy to be used based on the characteristics of the process;

2. the development of a user friendly interactive tool with graphical user interface for control design, simulation, and analysis of closed-loop systems, considering PID, DTC, and MPC controllers for processes with industrial characteristics;

3. a formulation of a PID tuning with a governor-like approach based on the GPC method for short control horizons, which is capable of providing optimal performance considering processes with input constraints, without the need for an online optimization procedure.

1.1 OBJECTIVES

1.1.1 General objective

The main objective of this work is to investigate control system approaches applied to industrial systems with characteristics such as dead time, constraints, modeling errors and measurement noise, and to develop practical solutions, tools, and insights based on common challenges and typical control structures used in this context. The main contributions of this research aim to facilitate the analysis and selection of the most convenient control structure

based on process characteristics, providing valuable guidance for control system design and implementation in industrial applications.

1.1.2 Specific objectives

The specific objectives of this thesis comprehend:

- Conduct a state-of-the-art survey of control structures used in processes with dead time and constraints;
- Provide insights for control theorists and industrial practitioners in selecting the most suitable control structure between PID, DTC or MPC controllers, based on process characteristics and available hardware;
- Develop a user-friendly simulation and analysis tool with graphical interface capable of simulating and comparing the performance and robustness of different control structures in various process scenarios;
- Formalize a simple and practical approach for fast computation of MPC with input constraint handling, considering short control horizons;

1.2 MAIN CONTRIBUTIONS AND PUBLISHED PAPERS

A total of five papers directly related to the doctoral research were produced. Three of them were published and presented at conferences, with the first one at the 13th Brazilian Symposium on Intelligent Automation (SBAI), the second at the 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control, and the third at the 21st IFAC World Congress. The fourth paper was published in the indexed journal ISA Transactions and won the Best Paper Award for the year 2020 among 450 other published articles. Lastly, the fifth paper has been submitted for revision to the journal ISA Transactions. The titles of the articles are listed below:

- Análise experimental de um GPC e um PID com anti-windup aplicados no controle de potência de uma ducha (SILVA; FLESCH; NORMEY-RICO,2017). 13th Brazilian Symposium on Intelligent Automation (SBAI 2017).
- Analysis of anti-windup techniques in PID control of processes with measurement noise (SILVA; FLESCH; NORMEY-RICO,2018). 3rd IFAC Conference on Advances in Proportional-Integral-Derivative (IFAC PID 2018).
- CSPS: an interactive tool for control design and analysis of processes with industrial characteristics (SILVA; FLESCH; NORMEY-RICO,2020). 21st IFAC World Congress (IFAC WC 2020).

- Controlling industrial dead-time systems: When to use a PID or an advanced controller (SILVA; FLESCH; NORMEY-RICO,2020). ISA Transactions (2020).
- PID design method based on GPC with input constraints handling (SILVA; FLESCH; NORMEY-RICO,2023). ISA Transactions (submitted for revision).

1.3 THESIS STRUCTURE

The manuscript is structured as a collection of papers. Introductions and literature reviews were omitted in each one of these paper-chapters to avoid content repetition for the reader. In Chapter 2 a comprehensive and in-depth literature review and theoretical foundations are presented, covering DTC and MPC strategies, as well as anti-windup approaches for constraint handling. The paper presented in Chapter 3 provides a comparative analysis of PID, DTC, and MPC strategies for controlling SISO processes with dead time, considering common industrial characteristics such as noisy measurements and modeling errors. In the paper presented in Chapter 4, a user-friendly simulation and analysis tool is introduced, capable of validating and comparing the performance and robustness of the three most widely used control structures in industrial applications: PID, DTC, and MPC, across different process scenarios. The paper in Chapter 5 presents a PID tuning approach that incorporates an additional structure designed to control first or second-order stable, integrative, or unstable processes subject to input constraints. Finally, Chapter 6 provides a general conclusion of the thesis.

2 LITERATURE REVIEW

In this chapter, a comprehensive literature review is presented, focusing on essential concepts related to dead-time compensators, specifically the classical structure of the SP and one of its primary extensions. Furthermore, it explores the objectives, capabilities, and formulation of MPC methods in the context of dealing with dead-time and constraints. Additionally, practical methods and structures are discussed to effectively handle input constraints, including anti-windup and governor methods. The chapter concludes by introducing performance and robustness indices used to analyze the closed-loop systems considered in this work.

2.1 DEAD-TIME COMPENSATORS

Dead-time compensators emerged intending to improve the closed-loop behavior of processes that present dead time in their dynamics. One of the main features of DTC strategies is that they make use of the process model to compute the control action, using the explicit information of the dead time in the control structure in order to obtain a dynamic performance considerably better when compared to classical strategies, such as PID control, for delay-dominant processes[1] (RIVERA; MORARI; SKOGESTAD, 1986; JEROME; RAY, 1986; NORMEY-RICO; CAMACHO, 2007). The following sections present a discussion regarding one of the most widely used DTC strategies in practical applications, along with one of its extensions, which enables its implementation for processes with varying dynamics. Additionally, a PID tuning method based on DTC ideas is presented.

2.1.1 Smith predictor

One of the first DTC approaches was presented in 1957 and became known as the Smith predictor (SP) (SMITH, 1957). This method is widely used in industry to deal with the dead-time problem, because it presents good performance when used to control open-loop stable delay-dominant processes (NORMEY-RICO; CAMACHO, 2007; SILVA; FLESCH; NORMEY-RICO, 2020). This control strategy uses the process model without the dead time in order to anticipate the process output to the controller avoiding the need to wait for the dead time to pass. Figure 1 presents the structure of the strategy, where $P_n(s) = G_n(s)e^{-L_n s}$ is the nominal process model, $G_n(s)$ is the fast model, L_n is the nominal dead time, $C_{\mathrm{SP}}(s)$ is the primary controller, $P(s) = G(s)e^{-Ls}$ is the real plant, $r(t)$ is the reference signal, $e(t)$ is the error signal, $u(t)$ is the control signal, $q(t)$ is an input disturbance, $\hat{y}(t)$ is the model output, $e_p(t)$ is the error between the actual output and the model output (prediction error), and $y_p(t)$ is the fast model output added to the prediction error $e_p(t)$.

[1] A processes is delay-dominant when $L_n \gg \tau$, where L_n is the nominal dead time and τ is the process time constant (NORMEY-RICO; CAMACHO, 2007)

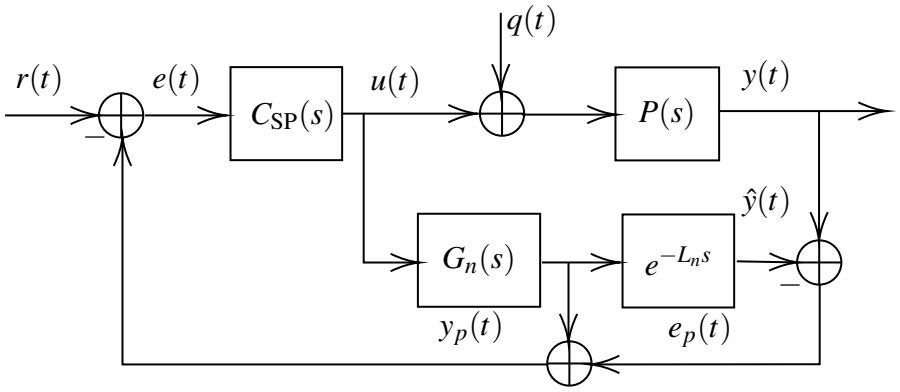


Figure 1 – Smith predictor structure

This structure is capable of predicting the expected behavior of the process output after the dead time has passed, $y_p(t)$, by computing the error between the actual output, $y(t)$, and the model output, $\hat{y}(t)$, and adding this value with the output of the fast model. Thus, considering the nominal case ($P_n(s) = P(s)$), the controller, $C(s)$, is able to compute the appropriate control action as if the system had no dead time (SMITH, 1957). This procedure makes it possible to design the primary controller considering the process model without the dead time, i.e. $G_n(s)$ (NORMEY-RICO; CAMACHO, 2007).

Considering the nominal case, SP presents three important features (NORMEY-RICO; CAMACHO, 2007), as detailed below.

1. The dead time is eliminated from the closed-loop characteristic equation. This characteristic can be observed by analyzing the closed-loop transfer function from the reference to the system output, $H_{yr}(s)$, given by

$$H_{yr}(s) = \frac{Y(s)}{R(s)} = \frac{C_{\mathrm{SP}}(s)P(s)}{1 + C_{\mathrm{SP}}(s)[P(s) - P_n(s) + G_n(s)]} \tag{2.1}$$

   where $Y(s) = \mathscr{L}\{y(t)\}$, $R(s) = \mathscr{L}\{r(t)\}$ and $\mathscr{L}$ is the Laplace transform.

   Considering the nominal case, it is possible to rewrite Equation (2.1) as

$$Y(s) = \frac{C_{\mathrm{SP}}(s)G_n(s)}{1 + C_{\mathrm{SP}}(s)G_n(s)} e^{-L_n s} R(s). \tag{2.2}$$

   Therefore, it is possible to observe that in Equation (2.2) the characteristic equation of the process, i.e. $1 + C_{\mathrm{SP}}(s)G_n(s) = 0$, does not present dead time.

2. In the nominal case, without disturbances, the feedback signal $y_p(t)$ computed by the SP anticipates the process output by L_n time instants ahead. This can be seen from the block diagram in Figure 1, which, when considering the nominal case and also considering no disturbances ($q(t) = 0$), results in a signal $e_p(t) = 0$, making the signal $y_p(s)$ exactly the same as the process output without dead time. Therefore, it is possible to state that signal $y_p(t)$ anticipates the behavior of the actual process output.

Another way to observe this characteristic is from the analysis of Equation (2.1) and the equation that relates the feedback signal $y_p(t)$ and the reference $r(t)$, given by

$$Y_p(s) = \frac{C_{\text{SP}}(s)G_n(s)}{1 + C_{\text{SP}}(s)G_n(s)}R(s), \tag{2.3}$$

where $Y_p(s) = \mathscr{L}\left\{y_p(t)\right\}$. Comparing equations (2.2) and (2.3) results in

$$Y(s) = Y_p(s)e^{-L_n s}. \tag{2.4}$$

Equation (2.4) can be represented in time-domain as

$$y_p(t) = y(t + L_n), \tag{2.5}$$

where $y(t + L_n)$ represents the process output L_n time-instants ahead.

3. SP is able to implicitly factorize the process into two parts: the fast model, $G_n(s)$, which in some cases may be invertible, and $e^{-L_n(s)}$, which is a non-invertible portion representing the dead time of the process.

Although the advantages of SP over other strategies that do not compensate the dead time, this structure has two limitations (NORMEY-RICO; CAMACHO, 2007): it is not possible to eliminate the open-loop dynamics from the disturbance response; SP can only be used to control stable processes.

These two limitations can be observed from the analysis of the transfer function that relates the process output, $y(t)$, with the disturbance, $q(t)$, which is given by

$$H_{yq}(s) = \frac{Y(s)}{Q(s)} = P_n(s)\left[1 - \frac{C_{\text{SP}}(s)P_n(s)}{1 + C_{\text{SP}}(s)G_n(s)}\right]. \tag{2.6}$$

It is possible to observe in Equation (2.6) the presence of the open-loop poles of the process model, which makes the SP structure not able to reject the disturbances faster than the open-loop dynamics of the process as it is not possible to tune $C_{\text{SP}}(s)$ to eliminate these poles from $H_{yq}(s)$. Also, if the process is unstable, $H_{yq}(s)$ will also be unstable, which makes impossible the use of SP to control this type of process (NORMEY-RICO; CAMACHO, 2009).

Several methods to overcome the problems presented by SP are presented in the literature (MAJHI; ATHERTON, 1998; PANDA; HUNG; YU, 2006; SANZ; GARCÍA; ALBERTOS, 2018). One of the methods that incorporates a fairly widespread solution to the issues presented by the SP is the one proposed in Normey-Rico and Camacho (2009), that consists of a unified structure known as filtered Smith predictor (FSP). This structure makes it possible to control stable, integrating, and unstable processes. In addition, this method enables the adjustment of a new parameter, which can be used to enhance the robustness of the system or improve its disturbance rejection performance in relation to the response of the open-loop system.

2.1.2 Filtered Smith predictor

The FSP strategy uses the same structure as the conventional SP with the addition of a unity static gain filter in the feedback of the system[2]. The FSP implementation presented in this chapter is for the discrete-time case, since this structure is easier to implement than the continuous-time one. However, the structure for the continuous case is equivalent, except for the output signal sampler and the zero-order hold (ZOH). More details of the implementation for the continuous case can be seen in (NORMEY-RICO; CAMACHO, 2009).

The discrete-time structure of the FSP can be seen in Figure 2, where d_n is the discrete dead time, $P_n(z) = G_n(z)z^{-d_n}$ is the process model discretized with a ZOH, $G_n(z)$ is the fast model of the process, $F(z)$ is the reference filter, $F_r(z)$ is the predictor filter, which allows to improve the properties of the original SP, $C_{\text{FSP}}(z)$ is the primary controller, and $P(s)$ is the plant. The discrete signals are represented as function of variable k, which is multiple of the sampling time T_s. In Figure 2, $r(k)$ is the reference, $e(k)$ is the error, $u(k)$ is the control action, $y(k)$ is the process output, $\hat{y}(k)$ is the prediction of the process output, $e_p(k)$ is the prediction error, $y_p(k)$ is the prediction of the output plus the filtered prediction error, $q(t)$ is the load disturbance, and $n(t)$ is the output disturbance (or it can also represent measurement noise).

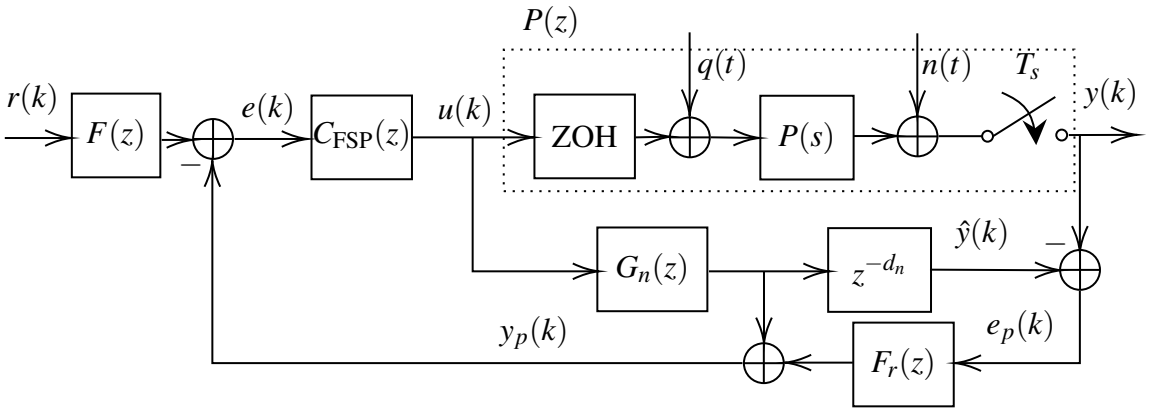


Figure 2 – Filtered Smith predictor structure

The original formulation of FSP, presented in Normey-Rico, Bordons and Camacho (1997), aimed to design the filter $F_r(z)$ in order to attenuate possible oscillations in the process output caused by modeling errors. However, in Normey-Rico and Camacho (2009), it was shown that, depending on the design of the filter $F_r(z)$, it is possible to obtain an improvement in the performance of the disturbance rejection or in the robustness of the system without affecting the nominal setpoint response. Furthermore, when properly designed, the addition of the filter turns the system internally stable, thus making it possible to control unstable or integrating plants, which is not possible using the original structure of SP (NORMEY-RICO; CAMACHO, 2009). For unstable processes, the structure shown in Figure 2 cannot be used for implemen-

[2] It is possible to consider a filter with a different static gain, depending on the tuning of the primary controller (TORRICO et al., 2013)

tation, because in this case, the unstable modes of the fast model generate internal instability. Therefore, for these cases, the structure shown in the Figure 3 is used, in which

$$S(z) = G_n(z)\left[1 - F_r(z)z^{-d_n}\right] \tag{2.7}$$

is a stable transfer function, since $F_r(z)$ is designed so that the zeros of $\left[1 - F_r(z)z^{-d_n}\right]$ cancel the poles of the model $G_n(z)$ that lie on or outside the unit circle. Furthermore, in the stable case, $F_r(z)$ can be tuned to eliminate the slow poles of $G_n(z)$. This tuning enhances the disturbance rejection response by ensuring that the slow poles do not appear in the closed-loop system.

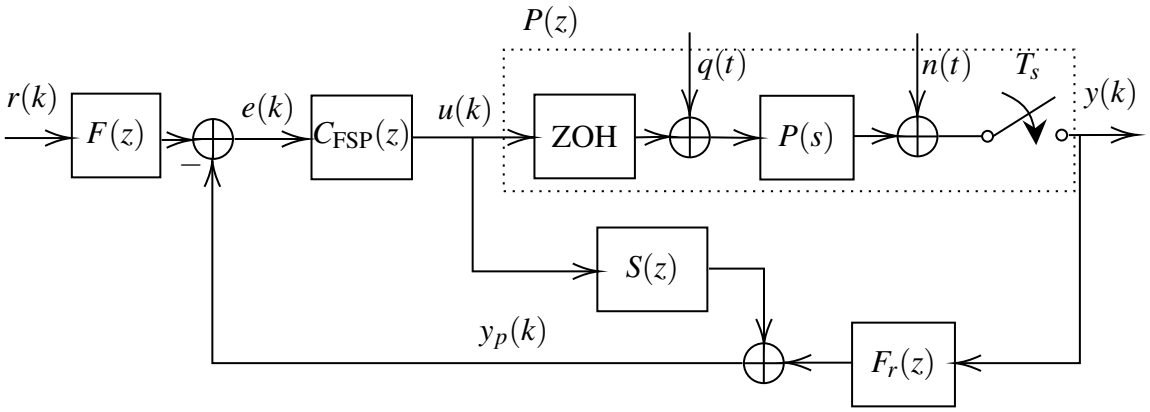


Figure 3 – Modified filtered Smith predictor structure for practical implementation

2.1.3 PID tuning based on FSP

This section presents a PID tuning, originally proposed in Normey-Rico and Guzmán (2013), for first-order plus dead time (FOPDT), integrating plus dead time (IPDT), or unstable first-order plus dead time (UFOPDT) processes, based on a low-frequency approximation of the FSP.

The main idea of this approach is based on two steps: firstly, the controller is designed based on the available first-order model and, then, the equivalent structure of the FSP is approximated by a PID. The PID formulation is based on the continuous-time FSP structure, shown in Figure 4, which considers the same elements shown in Figure 2.

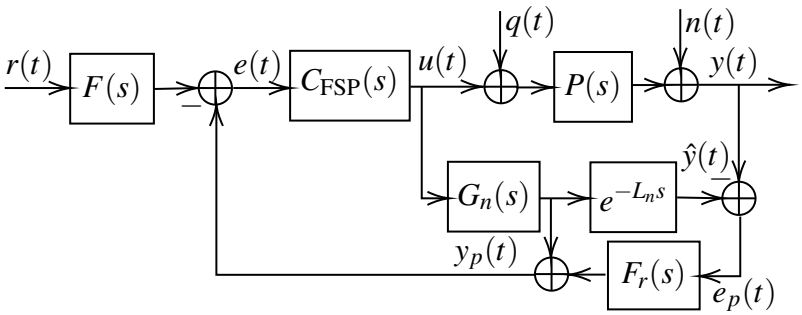


Figure 4 – Filtered Smith predictor structure

The primary controller, $C_{\text{FSP}}(s)$, used to tune the FSP is a PI controller with the following form

$$C_{\text{FSP}}(s) = \frac{K(s\tau_i + 1)}{s\tau_i}, \tag{2.8}$$

where K and τ_i are the proportional gain and integral-time constant, respectively.

In the structure, $G_n(s)$ can be described in three ways: $G_n(s) = K_G/(sT+1)$, for the stable case; $G_n(s) = K_G/s$, for the integrating case; and $G_n(s) = K_G/(sT-1)$ for the unstable case. In all three cases, the controller, $C_{\text{FSP}}(s)$, is tuned to obtain a closed-loop system with one or two poles at $s = -1/T_0$. Therefore, the parameters of the PI controller must be: $\tau_i = T$, $K = T/(T_0 K_G)$, for the stable case; $K = 1/(T_0 K_G)$ (proportional controller), for the integrating case; and $\tau_i = T_0/(2 + T_0/T)$, $K = (T_0 + 2T)/(T_0 K_G)$, for the unstable case. The term T_0 is the tuning parameter that defines the closed-loop time constant of the system and it is also used to define a trade-off between disturbance rejection and robustness.

The design of the filter $F_r(s)$ depends on the process model which is used. In all cases $F_r(s)$ can be designed to improve the robustness or the disturbance rejection response of the system. However, it is important to note that these two specifications cannot be achieved simultaneously and the tuning has to achieve a trade-off between robustness and disturbance rejection performance (NORMEY-RICO; CAMACHO, 2009). Additionally, for plants modeled as IPDT or as UFOPDT models, the filter $F_r(s)$ must be designed to avoid that the open-loop pole of the model appears in the disturbance response. Hence, the filter must be tuned according to Equation (2.9), for stable or integrating processes, and to Equation (2.10), for unstable processes.

$$F_{r_1}(s) = \frac{1 + s\beta}{1 + sT_0} \tag{2.9}$$

$$F_{r_2}(s) = \frac{1 + s\beta}{1 + sT_0(2 + T_0/T)} \tag{2.10}$$

In addition, a reference filter given by

$$F(s) = \frac{1 + sT_r}{1 + s\tau_i} \tag{2.11}$$

can be used to eliminate the effect of the zero of the controller considering the unstable case. The parameter T_r of the filter is capable to regulate the overshoot of the reference tracking response. With these settings, the FSP can improve the disturbance response when compared to the original SP, and also allows the control of integrative and unstable systems.

The equivalent structure of FSP is shown in Figure 5, where $C_{\text{eq}}(s)$ and $F_{\text{eq}}(s)$ can be computed as

$$\begin{aligned} C_{\text{eq}}(s) &= \frac{C_{\text{FSP}}(s)F_r(s)}{1 + C_{\text{FSP}}(s)G_n(s)[1 - e^{-L_n s}F_r(s)]}, \\ F_{\text{eq}}(s) &= \frac{F(s)}{F_r(s)}. \end{aligned} \tag{2.12}$$

It is important to note that the structure in Figure 5 is tipically implemented in the discrete-time domain. Also, the equivalent 2DOF FSP representation is used as basis for the PID approximation. Thus, the next step of the PID approximation is to obtain the equations of the equivalent

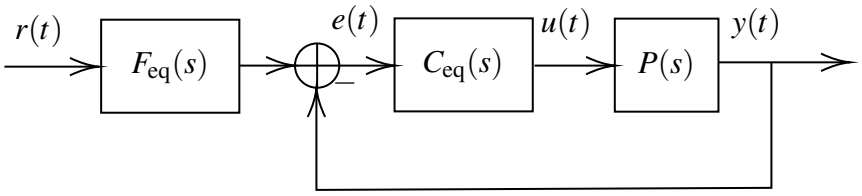


Figure 5 – Equivalent structure of the filtered Smith predictor

controllers for the stable, integrating, and unstable cases, from the parameters of the first-order model and the robustness filter $F_r(s)$. The equivalent controllers can be obtained from Equations (2.13), (2.14), and (2.15), where $C_{\text{eq}_1}(s)$, $C_{\text{eq}_2}(s)$ and $C_{\text{eq}_3}(s)$ represent, respectively, the equivalent FSP controllers for the stable, integrating, and unstable cases and the parameter β must be calculated in order to eliminate the zero of the equivalent controller, which is allocated at $s = -1/T$ in the stable case, at $s = 0$ for the integrating case, and $s = 1/T$ for the unstable case.

$$C_{\text{eq}_1}(s) = \frac{K(1+\beta s)(1+sT)}{s\tau_i(1+sT) + KK_G[1+s\tau_i - e^{-L_n s}(1+s\beta)]} \tag{2.13}$$

$$C_{\text{eq}_2}(s) = \frac{K(1+\beta s)s}{s(1+sT_0) + KK_G[1+sT_0 - e^{-L_n s}(1+\beta s)]} \tag{2.14}$$

$$C_{\text{eq}_3}(s) = \frac{K(1+\beta s)(1-sT)}{s\tau_i(1-sT) + KK_G[1+s\tau_i - e^{-L_n s}(1+s\beta)]} \tag{2.15}$$

For the computation of the PID controller, $C_{\text{PID}}(s)$, which approximates the FSP equivalent controller $C_{\text{eq}}(s)$, a first-order Padé approximation for the dead time is used, that is

$$e^{-L_n s} = \frac{1-0.5L_n s}{1+0.5L_n s}. \tag{2.16}$$

Finally, it is possible to find an expression that approximates the FSP by a PID series controller given by

$$C_{\text{PID}}(s) = \frac{k_c(1+sT_i)(1+sT_d)}{sT_i(1+s\alpha T_d)}, \tag{2.17}$$

where $T_d = 0.5L_n$. The parameters k_c, T_i and α can be calculated depending on the model considered. For the stable case

$$\begin{aligned} T_i &= T\left[1 - \frac{(2T-L_n)(T-T_0)^2}{(2T+L_n)T^2}\right], \\ k_c &= \frac{T_i}{K_G(L_n+2T_0-T_i)}, \\ \alpha &= \frac{T_0}{L_n+T_0+\frac{L_nT_i}{2T_0}-\frac{T_0L_n}{2T}}, \end{aligned} \tag{2.18}$$

for the integrating case

$$\begin{aligned} T_i &= 2T_0 + L_n, \\ k_c &= \frac{2 + \frac{L_n}{T_0}}{K_G \delta}, \\ \alpha &= \frac{T_0}{\delta}, \\ \delta &= T_0 + 0{,}5 L_n (4 + \frac{L_n}{T_0}), \end{aligned} \tag{2.19}$$

and for the unstable case

$$\begin{aligned} T_i &= \frac{T_0(2T + T_0)(2 + L_n/T) + 2L_n T}{2T - L_n}, \\ k_c &= \frac{T_i T}{K_G \delta}, \\ \alpha &= \frac{T_0^2}{\delta}, \\ \delta &= T_0^2 - T(L_n + T_0(2 + T_0/T) - T_i). \end{aligned} \tag{2.20}$$

In order to show that the PID controller approach adequately approximates a FSP, Figure 6 presents the frequency analysis of the magnitude of the controllers, considering examples for the particular cases of a stable ($P_s(s) = e^{-0.5s}/(s+1)$), an integrative ($P_i(s) = e^{-s}/s$), and an unstable process ($P_u(s) = e^{-0.4s}/(s-1)$). As can be seen in Figure 6, the PID controller, $C_{\text{PID}}(s)$,

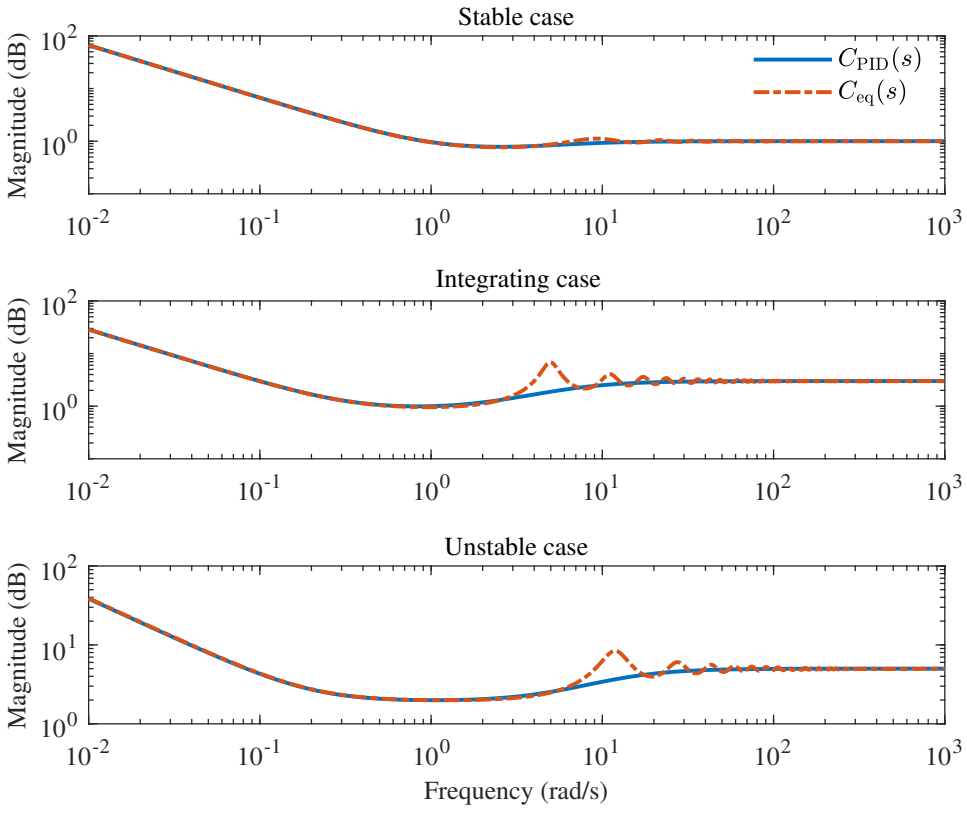


Figure 6 – Comparative analysis of the frequency response of the PID approximation and of the FSP for the stable, integrating, and unstable cases

presents a good approximation for the FSP, $C_{\text{eq}}(s)$, over a wide range of frequencies. This PID design will be used in Chapter 3 to perform a study and comparisons between PID, DTC and MPC strategies.

2.2 MODEL PREDICTIVE CONTROL

The term model predictive control is not related to one specific control method, but a wide range of strategies that make use of the process model to predict future outputs and use this information to compute an optimal control action by solving an optimization problem (CAMACHO; BORDONS, 2013). The main ideas of the MPC methods include (NORMEY-RICO; CAMACHO, 2007):

- an explicit use of a process model to compute future output predictions over a prediction horizon (N);
- a computation of a sequence of control actions by solving an optimization problem considering a specified control horizon (N_u) ;
- the application of only the first value of the computed control actions to the process and the repetition of the entire procedure in the next sampling period, as new information on the current state of the plant will be available.

The main difference between the various MPC algorithms presented in the literature is mainly related to the cost function to be minimized and in the model used to represent the process and disturbances (CAMACHO; BORDONS, 2013). MPC strategies present a series of benefits when compared to classical control methods. The main benefits include (NORMEY-RICO; CAMACHO, 2007):

- the MPC theory is relatively easy to understand, making it a good choice for users who do not have extensive knowledge of control systems;
- it can be used to control a wide range of processes, from simple to complex ones, such as dead-time, nonminimum-phase, and unstable processes;
- it naturally compensates dead time and can also include feedforward compensation to handle measurable disturbances;
- the process constraints can be included in the design process, making it useful for controlling systems with physical limitations or boundaries;
- when future references are known, MPC is very useful for controlling the system to meet those future goals.

The performance of MPC strategies is highly dependent on the quality of the process model, which can be a disadvantage in processes that are difficult to estimate or time-varying. Furthermore, according to Camacho and Bordons (2013), a disadvantage of the MPC in relation to the classic PID controller is that its implementation is more complex. However, in cases where there are no constraints and the process dynamics does not change with time, an explicit derivation of the controller can be performed offline.

The following sections provide an overview of the fundamental components of typical MPC methods, including examples of controllers in this family, and an in-depth formulation of GPC for both unconstrained and constrained cases. Additionally, a comparative analysis of the relationship between GPC and FSP is presented. This analysis establishes the foundation for the studies presented in Chapters 3 to 5.

2.2.1 Objective and ideia of MPC

The main objectives of the MPC correspond to (QIN; BADGWELL, 2003):

- avoid violations of input and output constraints;
- take the controlled variable to an optimal reference value;
- prevent sudden actions on the manipulated variable.

Figure 7 shows the MPC idea considering a SISO system. The future output predictions are computed using the process model. These predictions, $\hat{y}(k+j|k)$, for $j = 1,\ldots,N$, where N is known as prediction horizon, depend on the past inputs and outputs of the plant and the future control actions, $u(k+j)$ for $j = 0,\ldots,N_u-1$, where N_u is known as control horizon[3].

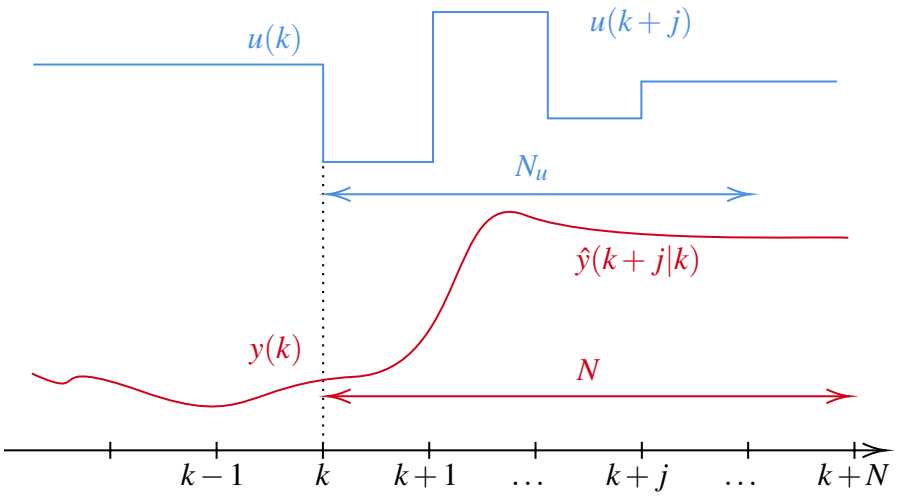


Figure 7 – MPC idea (CAMACHO; BORDONS, 2013)

[3] The notation $\hat{y}(k+j|k)$ refers to the predicted value of the process variable at instant $k+j$ computed at instant k

MPC computes a set of control actions based on the minimization of an objective function that typically takes into account future prediction errors (between future references and future output predictions) and also penalizes the control increments, in order to obtain smooth changes in control actions. The justification for the MPC method to apply only the first control action is to avoid predictions and control actions based on outdated information, which can affect the robustness of the controller in the presence of unmeasured disturbances and modeling errors (SEBORG; EDGAR; MELLICHAMP, 2006).

A basic structure of the MPC method is shown in Figure 8. The process model is used to predict future plant outputs using current and past values, as well as control actions computed by the optimizer. These control actions are computed based on the cost function and taking into account process constraints and the reference signal to be tracked.

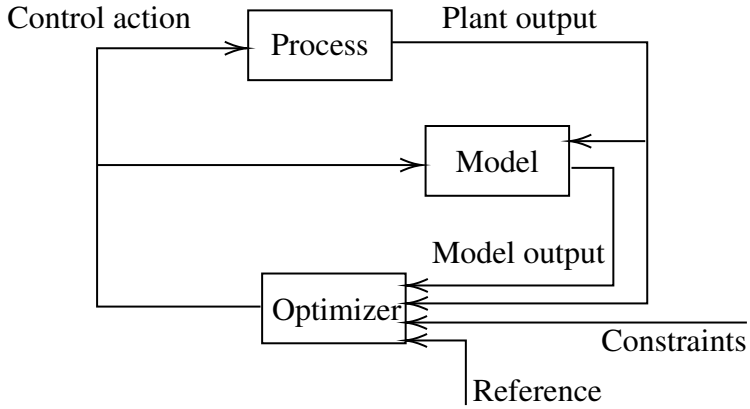


Figure 8 – MPC structure (NORMEY-RICO; CAMACHO, 2007)

2.2.2 Prediction models

According to Wang (2009), there are three main types of models that MPC can use to compute future process output predictions using linear models: step response, transfer function, and state-space models. The dynamic matrix control (DMC), proposed by Cutler and Ramaker (1980), which was one of the first MPC methods published in the literature, and quadratic dynamic matrix control (QDMC) (GARCIA; MORSHEDI, 1986), employ the coefficients of the step response for their formulation. The main disadvantage of these methods, in their original algorithms, is that they can only be applied to stable processes, as the step response is unable to model unstable plants (CAMACHO; BORDONS, 2013). On the other hand, the MPC formulations that use transfer function or state-space models, such as Shell multivariable optimization control (SMOC) (MARQUIS; BROUSTAIL, 1988), generalized predictive control (GPC) (CLARKE; MOHTADI; TUFFS, 1987), extended horizon adaptive controller (EHAC), (YDSTIE, 1984), unified predictive Control (UPC) (SOETERBOEK, 1992) and receding horizon tracking control (RHTC) (KWON; BYUN, 1989) are capable to control unstable plants.

In this work, the transfer function model has been chosen as model representation because it is believed to be the approach that is most compact when compared to step-response

models, most easily connected with models used in industry and with DTC and PID methods, given the similarity in model representation.

A discrete-time transfer function that represents a SISO linear process is given by:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{N(z)}{D(z)}, \tag{2.21}$$

where $Y(z) = \mathscr{Z}\{y(k)\}$, $U(z) = \mathscr{Z}\{u(k)\}$, $\mathscr{Z}$ is the Z-transform, and $N(z)$ and $D(z)$ are polynomials.

According to Normey-Rico and Camacho (2007), it is typically assumed that the process does not have an instantaneous response. Thus, it is possible to write Equation (2.21) as a function of the backward shift operator z^{-1}, relating the system input, $u(k)$, and the system output, $y(k)$, as follows:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k-1), \tag{2.22}$$

where $A(z^{-1})$ and $B(z^{-1})$ are polynomials given by

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_a} z^{-n_a}, \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{n_b} z^{-n_b}. \end{aligned} \tag{2.23}$$

In this way, it is possible to represent the process future output prediction as

$$\hat{y}(k+j|k) = \frac{B(z^{-1})}{A(z^{-1})}u(k-1+j), \tag{2.24}$$

with $j = 1,2,\ldots N$. The major advantage of this representation over step or impulse response coefficients is that it can be used to represent unstable processes (CAMACHO; BORDONS, 2013).

The choice of a model to represent the system disturbances is also very important in the formulation of MPC methods. A widely used model is the controlled auto-regressive integrated moving average (CARIMA) (GOODWIN-SIN, 1984). According to Clarke, Mohtadi and Tuffs (1987), this model is highly suitable for industrial applications, where the disturbance is usually non-stationary. The CARIMA model is discussed in detail in Section 2.2.5, where the GPC formulation, which uses this type of model to compute the output predictions, is presented.

2.2.3 Free and forced responses

Two concepts widely used in MPC theory for linear models are the concepts of free response and forced response. The goal is to represent the process response as the sum of two signals

$$y(k) = y_f(k) + y_c(k). \tag{2.25}$$

The free response, $y_f(k)$, represents the prediction of the future outputs of the process if the manipulated variable of the system remains constant for future values, that is, $\Delta u(k+j) = 0$,

for $j = 1, \ldots$. The forced response of the system, $y_c(k)$, represents the prediction of the future outputs of the process taking into account only the future increments of the control action, that is $\Delta u(k+j)$ for $j = 0, \ldots$ (CAMACHO; BORDONS, 2013). Thus, it is possible to separate the part of the response that depends on future control actions from that which does not depend. In short, the free response can be seen as the evolution dynamics of the process given its current state and the forced response can be interpreted as the evolution of the output due to variations in the manipulated variable (WANG, 2009).

For linear processes, the predicted output can be calculated as the sum of the free response and the forced responses (principle of superposition). However, the concept of superposition cannot be applied to nonlinear processes (HABER; BARS; SCHMITZ, 2012).

2.2.4 Objective function

There are several proposed objective functions that can be used to obtain the optimal control signal for the process. In different types of MPC methods, the main objective of the system is to ensure that the predicted future output, $\hat{y}(k+j|k)$ for $j = 1, \ldots, N$, tracks a certain reference signal, $r(k+j)$ for $j = 1 \ldots N$, throughout the prediction horizon, while the control signal variation, $\Delta u(k+j)$ for $j = 0, \ldots, N_u - 1$, is as small as possible.

When a process has a dead time of d sampling periods in its dynamics, the system output will only be influenced by the control signal, $u(k)$, after $d+1$ sampling periods have passed. Thus, it makes no sense to choose a value of N that is smaller than or equal to d. Instead, it is recommended to use N_1 and N_2 as variables to define the beginning and end of the prediction horizon in the objective function. Specifically, we can define $N_1 = d+1$ and $N_2 = d+N$.

An objective function that meets the previously mentioned criteria can be mathematically expressed as a quadratic function given by

$$J_{\text{MPC}} = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(k+j|k) - r(k+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(k+j-1)]^2, \tag{2.26}$$

where N_1 and N_2 represent the beginning and end of the prediction horizon, N_u represents the control horizon, $\delta(j)$ is the weighting of the reference tracking error, and $\lambda(j)$ is the control signal weight. Usually, the choice of N for the definition of N_1 and N_2 depends on the dynamic characteristics of the process.

Another point worth mentioning is that multiplying a quadratic function by a constant does not change the result of the optimization process. Therefore, for SISO processes, when the weighting of the reference tracking error and control signal is constant throughout the horizon, it is possible to use one of them as tuning parameter and consider the other unitary, for example, if $\delta(j) = 1$ for $j = N_1, \ldots, N_2$ it is possible to use $\lambda(j) = \lambda$ for $j = N_1, \ldots, N_2$ as a tuning parameter.

2.2.5 Generalized predictive control

The GPC method employs a model obtained from a transfer function to predict the future process outputs, and uses this information to compute an optimal control sequence by minimizing the objective function presented in Equation (2.26) (CLARKE; MOHTADI; TUFFS, 1987). This type of controller is capable of controlling simple systems, such as open-loop stable plants, or more complex systems, such as non-minimum phase, unstable or dead-time processes (CAMACHO; BORDONS, 2013). The GPC strategy uses the CARIMA model to compute future predictions of the process output. The CARIMA model is represented in Equation (2.27), where $u(k)$ and $y(k)$ represent the process input and output, respectively, $I(z^{-1})$ is a polynomial in the delay operator z^{-1} which represents the stochastic characteristic of the noise (usually considered equal to 1 if this characteristic is unknown), d is the dead time in samples, $\xi(k)$ is a zero-mean white noise, $\Delta = (1 - z^{-1})$, and $A(z^{-1})$ and $B(z^{-1})$ are polynomials in z^{-1}(CAMACHO; BORDONS, 2013).

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{I(z^{-1})\xi(k)}{\Delta} \tag{2.27}$$

For the case in which the stochastic characteristic of the model is unknown, i.e. when $I(z^{-1}) = 1$, the process output predictions for an instant $k+j$ can be computed from the process model (equation (2.24)) by multiplying both sides of the Equation by Δ, resulting in the equation

$$\tilde{A}(z^{-1})y(k+j) = B(z^{-1})\Delta u(k-d+j-1) + \xi(k+j), \tag{2.28}$$

where $\tilde{A}(z^{-1}) = A(z^{-1})\Delta$. Using the Diophantine equation, given by

$$1 = \tilde{A}(z^{-1})E_j(z^{-1}) + z^{-j}F_j(z^{-1}), \tag{2.29}$$

and multiplying both sides of Equation (2.28) by $E_j(z^{-1})$, we get

$$\tilde{A}(z^{-1})E_j(z^{-1})y(k+j) = E_j(z^{-1})\xi(k+j) + E_j(z^{-1})B(z^{-1})\Delta u(k-d+j-1). \tag{2.30}$$

Substituting (2.29) in Equation (2.30), and isolating $y(k+j)$ results in the following expression

$$y(k+j) = F_j(z^{-1})y(k) + E_j(z^{-1})B(z^{-1})\Delta u(k-d+j-1) + E_j(z^{-1})\xi(k+j). \tag{2.31}$$

As the degree of the polynomial E_j is $j-1$, the white noise terms $\xi(k+j)$ are all in the future, thus, the expected value of these terms, which is null, is used to perform the predictions. After all the modifications, the expression that represents the future output predictions is given by

$$\hat{y}(k+j|k) = G_j(z^{-1})\Delta u(k-d+j-1) + F_j(z^{-1})y(k), \tag{2.32}$$

where $\hat{y}(k+j|k)$ is the prediction at $k+j$ computed with the data up to k, and the term $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$ contains the values referred to the step response of the process model. Depending on the number of zeros of the process model, some coefficients which multiply the

past Δu values may appear, resulting from the multiplication $G_j(z^{-1})\Delta u(k+j-d-1)$. Considering $N_1 = d+1$ and $N_2 = d+N$, if the predictions for all instants of the horizon are grouped into a vector $\hat{\mathbf{y}}$, this vector can be written as

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{F}(z^{-1})y(k) + \mathbf{G}'(z^{-1})\Delta u(k-1), \tag{2.33}$$

where

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k+d+1|k) \\ \hat{y}(k+d+2|k) \\ \vdots \\ \hat{y}(k+d+N|k) \end{bmatrix}, \qquad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u} & g_{N_u-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \cdots & g_{N-N_u+1} \end{bmatrix},$$

$$\mathbf{G}'(z^{-1}) = \begin{bmatrix} (G_{d+1}(z^{-1}) - g_1)z \\ (G_{d+2}(z^{-1}) - g_1 - g_2 z^{-1})z^2 \\ \vdots \\ (G_{d+N}(z^{-1}) - g_1 - g_2 z^{-1} - \ldots - g_N z^{-(N-1)})z^N \end{bmatrix},$$

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{bmatrix},$$

where the terms of $\mathbf{G}$ are the coefficients of the step response of the model. The terms $\mathbf{G}'(z^{-1})$ and $\mathbf{F}(z^{-1})$ depend on past values and, therefore, are grouped under the term $\mathbf{f}$, resulting in

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}, \tag{2.34}$$

where $\mathbf{f}$ is known as free response, which can be obtained considering $\Delta u(k+j) = 0$, for $j = 0, \ldots, N_u - 1$, and the term $\mathbf{G}\Delta\mathbf{u}$ represents the forced response of the system.

By Substituting (2.34) in Equation (2.26) it is possible to represent the objective function as

$$J_{\text{MPC}} = (\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{r})^T \mathbf{Q}_\delta (\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{r}) + \mathbf{u}^T \mathbf{Q}_\lambda \mathbf{u}, \tag{2.35}$$

where $\mathbf{Q}_\delta$ is a diagonal matrix $N \times N$ containing the elements of $\delta(j)$, $\mathbf{Q}_\lambda$ is a diagonal matrix $N_u \times N_u$ containing the elements of $\lambda(j)$ and $\mathbf{r} = [r(k+N_1)\ r(k+N_1+1) \cdots r(k+N_2)]^T$ is a vector of dimension N with future reference signals.

Rewriting Equation (2.35) in the form of a quadratic programming (QP) problem results in

$$J_{\text{MPC}} = \frac{1}{2}\Delta\mathbf{u}^T\mathbf{H}\Delta\mathbf{u} + \mathbf{b}^T\Delta\mathbf{u} + f_0, \tag{2.36}$$

where

$$\begin{aligned} \mathbf{H} &= 2(\mathbf{G}^T\mathbf{Q}_\delta\mathbf{G} + \mathbf{Q}_\lambda), \\ \mathbf{b}^T &= 2(\mathbf{f}-\mathbf{r})^T\mathbf{Q}_\delta\mathbf{G}, \\ f_0 &= (\mathbf{f}-\mathbf{r})^T\mathbf{Q}_\delta(\mathbf{f}-\mathbf{r}). \end{aligned} \tag{2.37}$$

According to Clarke, Mohtadi and Tuffs (1987), the optimal solution, considering the unconstrained case, is linear and can be found analytically by applying the gradient of J_{MPC} and setting it equal to zero, which leads to

$$\Delta\mathbf{u} = -\mathbf{H}^{-1}\mathbf{b} = (\mathbf{G}^T\mathbf{Q}_\delta\mathbf{G} + \mathbf{Q}_\lambda)^{-1}\mathbf{G}^T\mathbf{Q}_\delta(\mathbf{r}-\mathbf{f}). \tag{2.38}$$

In GPC strategy, only the first control action computed is applied to the process and it can be obtained as

$$u(k) = u(k-1) + \Delta u(k). \tag{2.39}$$

Typically, the constraints used in MPC are related to the amplitude and variation of the control signal and the amplitude of the system output. The objective function that must be minimized, taking into account these constraints, can be mathematically represented as:

$$\begin{aligned} \min \quad & J_{\text{MPC}} \\ \text{s.t.} \quad & u_{\min} \le u(k+j) \le u_{\max} && \forall j, \\ & \Delta u_{\min} \le \Delta u(k+j) \le \Delta u_{\max} && \forall j, \\ & y_{\min} \le y(k+j) \le y_{\max} && \forall j, \end{aligned} \tag{2.40}$$

where $u_{\min}$ and $u_{\max}$ are the minimum and maximum limits of the control action, $\Delta u_{\min}$ and $\Delta u_{\max}$ are the minimum and maximum limits of the control action increment, and $y_{\min}$ and $y_{\max}$ are the minimum and maximum limits of the process output.

Considering a prediction horizon of N and a control horizon of N_u, these constraints can be rewritten as function of vector $\Delta\mathbf{u}$ as

$$\begin{aligned} &\mathbf{1}u_{\min} \le \mathbf{T}\Delta\mathbf{u} + u(k-1)\mathbf{1} \le \mathbf{1}u_{\max}, \\ &\mathbf{1}\Delta u_{\min} \le \Delta\mathbf{u} \le \mathbf{1}\Delta u_{\max}, \\ &\tilde{\mathbf{1}}y_{\min} \le \mathbf{G}\Delta\mathbf{u} + \mathbf{f} \le \tilde{\mathbf{1}}y_{\max}, \end{aligned} \tag{2.41}$$

where $\mathbf{1} \in \mathbb{R}^{N_u}$ and $\tilde{\mathbf{1}} \in \mathbb{R}^{N}$ are vectors of ones and $\mathbf{T} \in \mathbb{R}^{N_u \times N_u}$ is a lower triangular matrix in which all non-null elements are equal to one. Thus, the constraints described in Equation (2.41) can be represented in a simplified form as

$$\mathbf{A_c}\Delta\mathbf{u} \le \mathbf{b_c}, \tag{2.42}$$

where

$$\mathbf{A_c} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{T} \\ -\mathbf{T} \\ \mathbf{G} \\ -\mathbf{G} \end{bmatrix}, \mathbf{b_c} = \begin{bmatrix} \mathbf{1}\Delta u_{\max} \\ -\mathbf{1}\Delta u_{\min} \\ \mathbf{1}(u_{\max} - u(k-1)) \\ -\mathbf{1}(u_{\min} - u(k-1)) \\ \tilde{\mathbf{1}} y_{\max} - \mathbf{f} \\ -\tilde{\mathbf{1}} y_{\min} + \mathbf{f} \end{bmatrix}, \tag{2.43}$$

and $\mathbf{I} \in \mathbb{R}^{N_u \times N_u}$ is the identity matrix. Finally, to compute the optimal control action, the minimization of a quadratic function with affine constraints given by

$$\begin{aligned} &\min \quad J_{\text{MPC}} \\ &\text{s.t.} \quad \mathbf{A_c}\Delta\mathbf{u} \leq \mathbf{b_c}, \end{aligned} \tag{2.44}$$

with J_{MPC} defined in Equation (2.36), must be solved.

2.2.6 Relation between GPC and FSP

In Normey-Rico and Camacho (2007), it is shown that GPC, considering the unconstrained case, can be represented as an FSP with two degrees of freedom, similar to the structure shown in Figure 2, with a reference filter, $F(z)$, a primary controller, $C_{\text{FSP}}(z)$, and a predictor filter, $F_r(z)$, given by

$$C_{\text{FSP}}(z) = -\frac{l_{y_1} + \ldots + l_{y_{n_a+1}} z^{-n_a}}{(1-z^{-1})(1 - l_{u_1} z^{-1} - \ldots - l_{u_{n_b}} z^{-n_b})}, \tag{2.45}$$

$$F(z) = -\frac{v_1 z^{d+1} + \ldots + v_N z^{d+N}}{l_{y_1} + \ldots + l_{y_{n_a+1}} z^{-n_a}}, \tag{2.46}$$

$$F_r(z) = \frac{l_{y_1} F_d(z^{-1}) + \ldots + l_y n_a + 1 F_{d-n_a}(z^{-1})}{l_{y_1} + \ldots + l_{y_{n_a+1}} z^{-n_a}}, \tag{2.47}$$

where the coefficients $[l_{y_1}, \ldots l_{y_{n_a+1}}]$, $[l_{u_1}, \ldots, l_{u_{n_b}}]$, and $[v_1, \ldots, v_N]$ can be computed depending on the parameters of the process model, the prediction horizon, N, and the weight of the control effort λ. Details on how to obtain the parameters of the equivalent structure can be seen in Normey-Rico and Camacho (2007). This analysis shows that, with proper tuning, the FSP can deliver the same response as GPC, considering the unconstrained case, for any process with dead time. Furthermore, in Normey-Rico and Camacho (2007) it is shown that using the same formulation to tune $F(z)$ and $C_{\text{FSP}}(z)$, but considering a different filter $F_r(z)$, it is possible to improve the robustness of the controller or speed up the disturbance response. This approach became known in the literature as DTC-GPC, and presents the same nominal reference tracking response as the original GPC structure.

2.3 INPUT CONSTRAINT HANDLING TECHNIQUES

Figure 9 represents a discrete-time diagram with a saturation constraint, where $r(k)$ is the reference signal, $e(k)$ is the error, $u(k)$ is the control signal, $u_r(k)$ is the control signal applied to the plant, $y(k)$ is the process output, $C(z)$ is the controller, $P(z)$ is the discrete-time model of the plant, $u_{\min}$ and $u_{\max}$ are the minimum and maximum saturation values of the control signal.

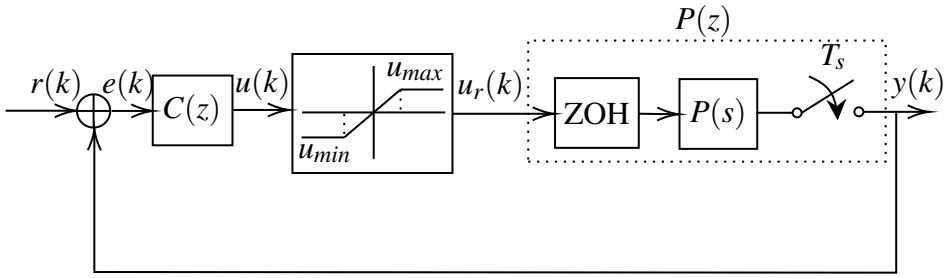


Figure 9 – Control structure of a discrete-time system with saturation constraints

An example in which actuator saturation occurs can be observed in processes in which it is necessary to control the power applied to a resistor. As it is not possible to apply powers with negative values, the system is limited to a minimum value of zero watts. The maximum possible value is limited by the resistor power or by the actuator voltage limitation. Thus, depending on the operating region in which the system operates, the calculated control signal may exceed the actuator limits and, from there, any change in the control signal does not result in a faster response of the system. This ends up generating a discrepancy between the value of the control signal computed by the controller and the value which is actually applied to the plant (ÅSTRÖM; HAGGLUND, 1995).

When saturation of the actuator occurs, the error signal remains large for a longer time than predicted by the linear model and, if the controller has any slow (integral action, for example) or unstable mode, the error signal is accumulated resulting in a very high value of the computed control action (HIPPE, 2006). The consequence is a control signal value which is in a region outside of the actuator operating range for a very long time, causing the controller to take some time to bring the control signal back to the linear region and correct the error. This phenomenon, known in the literature as windup, if it is not appropriately managed, can degrade the controller performance in relation to the linear performance. In order to reduce the unwanted effects caused by actuator saturation, several methods have been proposed in the literature over the years. One of the most simplified and practical approaches is known as AW.

Basically, AW methods consist of incorporating an auxiliary structure in the closed-loop system to maintain the output of the controller within the allowable limits of the actuator, thereby preventing the accumulation of error. One of the main advantages of this method is that it can be applied to systems without the need to modify the tuning of the controller that is already implemented and operating in the plant, making it possible to design the AW structure

a posteriori.

Another methods widely used in practice to avoid the violation of constraints are the governor approaches. These methods consist in a separate structure which is capable of modifying a state of the controller in order to reduce the effect in closed-loop caused by the actuator saturation.

From Sections 2.3.1 to 2.3.4 some of the main input constraints handling methods in the literature are presented.

2.3.1 Incremental algorithm

The incremental algorithm (or velocity algorithm) method consists in computing the increment of the control action, $\Delta u(k)$, at each sampling instant, and adding to the past control signal, $u(k-1)$, only the portion that does not saturate the system actuator (PENG; VRANCIC; HANUS, 1996). This strategy is capable of avoiding windup because in this method the integral action is outside of the controller (HIPPE, 2006).

In order to demonstrate the formulation of the incremental algorithm method, consider a controller described as function of the delay operator, z^{-1}, and given by

$$C(z^{-1}) = \frac{B_c(z^{-1})}{A_c(z^{-1})} = \frac{b_{c_0} + b_{c_1}z^{-1} + \ldots + b_{c_q}z^{-q}}{1 + a_{c_1}z^{-1} + \ldots + a_{c_s}z^{-s}}. \tag{2.48}$$

Considering $\Delta = 1 - z^{-1}$, it is possible to compute the increment and the control action, respectively, as

$$\Delta u(k) = b_{c_0}\Delta e(k) + b_{c_1}\Delta e(k-1) + \ldots + b_{c_q}\Delta e(k-q) - a_{c_1}\Delta u(k-1) - \ldots - a_{c_s}\Delta u(k-s). \tag{2.49}$$

$$u(k) = \Delta u(k) + u(k-1). \tag{2.50}$$

Considering the diagram shown in Figure 9 and Equation (2.50), if the control signal saturates, i.e. $\Delta u(k) + u(k-1) \neq u_r(k)$, the control increment must be recalculated so that

$$\Delta u(k) = u_r(k) - u(k-1), \tag{2.51}$$

which makes the current control action equal to the control action which is applied to the plant, that is $u(k) = u_r(k)$ (PENG; VRANCIC; HANUS, 1996).

2.3.2 Back-calculation

The back-calculation method, proposed by Fertik and Ross (1967), which was originally developed for parallel PID controllers, aims to avoid that the integral portion of the controller accumulates a high value when the controller is saturated, that is, $u(k) \neq u_r(k)$. Thus, the technique uses an extra feedback signal for the integral portion that is obtained by the difference

between the value of the controller output and the system input multiplied by a gain $1/T_t$, where T_t is known as the tracking-time gain. The gain T_t determines the speed with which the integral portion is reduced, forcing this term of the controller to be recomputed every time the control signal saturates (ÅSTRÖM; HAGGLUND, 1995).

The formulation of the back-calculation strategy can be obtained by rewriting the discrete-time parallel PID controller

$$C_{\text{PID}}(z^{-1}) = K_p + K_i T_s \frac{1}{1-z^{-1}} + \frac{K_d}{\alpha + T_s \frac{1}{1-z^{-1}}} \tag{2.52}$$

in the following form

$$u(k) = u_p(k) + u_i(k) + u_d(k), \tag{2.53}$$

where $u_p(k)$, $u_i(k)$, and $u_d(k)$ are, respectively, the proportional, integral, and derivative portions of the control signal. The three portions can be computed as

$$u_p(k) = K_p e(k), \tag{2.54}$$

$$u_i(k) = u_i(k-1) + K_i T_s e(k), \tag{2.55}$$

$$u_d(k) = \frac{K_d \left[e(k) - e(k-1)\right] + \alpha u_d(k-1)}{\alpha + T_s}. \tag{2.56}$$

When the control signal is saturated, the method adds to the integral portion a correction factor $e_{cf}(k) = u_r(k) - u(k)$ multiplied by the gain $1/T_t$. The new integral portion of the controller, $u_i^*(k)$, can then be computed as

$$u_i^*(k) = u_i(k-1) + \left[K_i e(k) + \frac{1}{T_t} e_{cf}(k)\right] T_s. \tag{2.57}$$

Finally, the new control signal, $u^*(k)$, can be calculated by substituting Equation (2.57) in Equation (2.53), resulting in

$$u^*(k) = u_p(k) + u_i^*(k) + u_d(k). \tag{2.58}$$

Figure 10 shows the back-calculation structure and Algorithm 1 demonstrates how to implement it.

2.3.3 Error recalculation

The work of Bruciapaglia and Apolônio (1986) proposes an AW strategy for PID controllers that aims to modify both the current control signal, in order to satisfy the constraints, and the current error signal, so that it has a value necessary to make the control signal equal to the saturation limit before leaving the linear region of operation. In this way, the consistency between the computed control signal, used in the recursive expression of the controller, and the

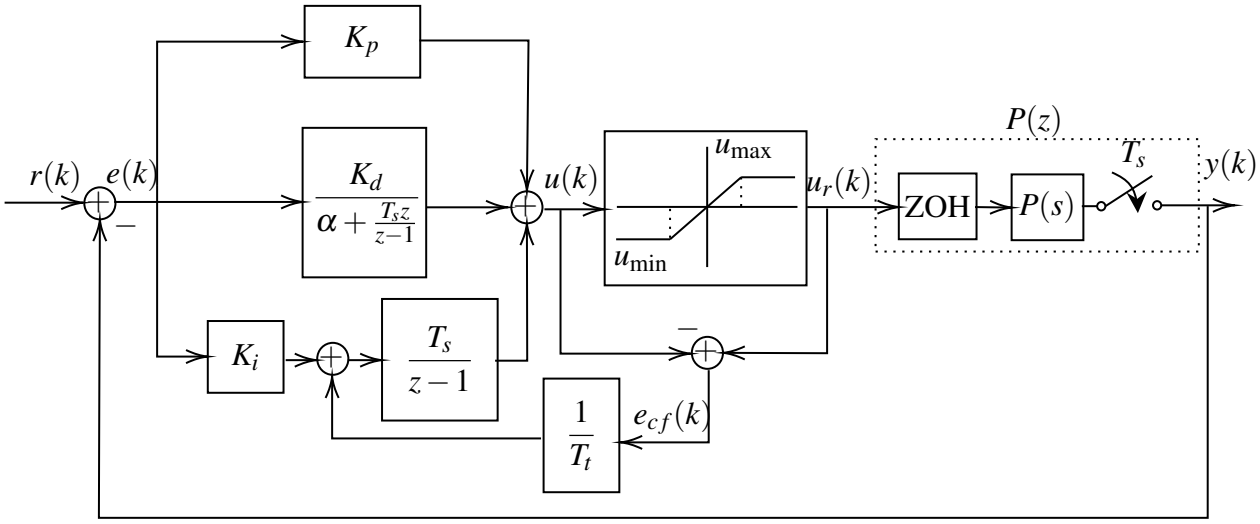


Figure 10 – Back-calculation structure

Algorithm 1: Anti-windup: back-calculation

```
initialize variables;
repeat
    measure plant output y(k);
    e(k) ← r(k) − y(k);
    compute control action u(k);
    if u(k) ≠ u_r(k) then
        e_cf(k) ← u_r(k) − u(k);
        u_i^*(k) ← u_i(k−1) + [K_i e(k) + (1/T_t) e_cf(k)];
        u(k) ← u_p(k) + u_i^*(k) + u_d(k);
    apply u(k) to the plant;
    update variables;
    k ← k + 1;
until controller is stopped;
```

signal which is applied to the plant is maintained. This technique holds some similarity with the idea of the EG approach, proposed by Kapasouris, Athans and Stein (1988). The main difference is that the EG approach uses an online optimizer to compute the optimal error signal based on an objective function, while the method proposed by Bruciapaglia and Apolônio (1986) does not require an optimizer.

A generalization of the approach of Bruciapaglia and Apolônio (1986) was proposed in the work of Flesch, R., Normey-Rico and Flesch, C. (2017) for controllers of any order and it is referred in this work by the name of error recalculation.

The method proposed by Flesch, R., Normey-Rico and Flesch, C. (2017) considers the discrete-time control problem presented in Figure 11, where $r(k)$ is the reference, $e(k)$ is the error, $u(k)$ is the desired control signal, $u_r(k)$ is the constrained control signal (plant input), $y^*(k)$ is the plant output, $n(k)$ is the noise, $y(k)$ is the measured output of the system, $C(z)$ is the controller, $\phi(\cdot)$ is the nonlinearity, and $P(z)$ is the plant.

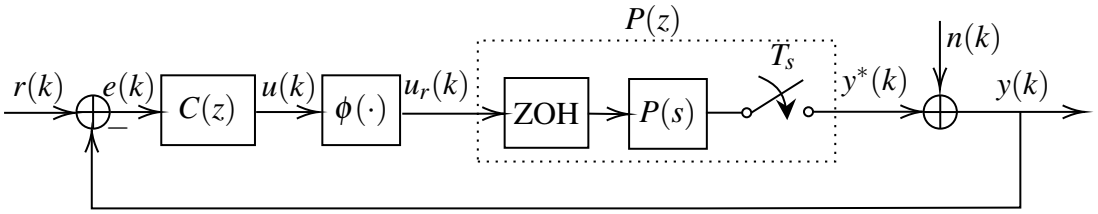


Figure 11 – Control structure of a discrete-time system with an input nonlinearity

The main idea of this approach is to modify both the current control signal and the current error signal to maintain the consistency between the control signal calculated by the controller and the input signal applied to the plant, i.e. $u(k) = u_r(k)$.

Based on Equation (2.48) it is possible to compute the current control signal as a function of the last s control signals, the last q errors, and the current error, resulting in

$$u(k) = b_{c_0}e(k) + b_{c_1}e(k-1) + \ldots + b_{c_q}e(k-q) - a_{c_1}u(k-1) - \ldots - a_{c_s}u(k-s). \tag{2.59}$$

Taking into consideration that, up to the previous instant, $k-1$, the desired control signals $u(k-1)\ldots u(k-s)$ are equal to the plant inputs $u_r(k-1)\ldots u_r(k-s)$ and the error signals $e(k-1)\ldots e(k-q)$ are equal to the error signals that would be expected for equalities $u(k-j) = u_r(k-j)$ with $j = 1\ldots s$ are satisfied, the following Equation is valid

$$u(k) = (1 - A_c(z^{-1}))u_r(k) + b_{c_0}e(k) + (B_c(z^{-1}) - b_{c_0})e^*(k), \tag{2.60}$$

where $e^*(k)$ represents the expected value for the current error so that the equality $u(k) = u_r(k)$ is respected (FLESCH, R.; NORMEY-RICO; FLESCH, C. 2017). The condition to implement the AW method is: if $u(k) \neq u_r(k)$, the value of $e(k)$ must be changed so that $u(k) = u_r(k)$. This can be expressed as

$$u_r(k) = (1 - A_c(z^{-1}))u_r(k) + b_{c_0}e^*(k) + (B_c(z^{-1}) - b_{c_0})e^*(k). \tag{2.61}$$

By subtracting Equation (2.61) from Equation (2.60), it is possible to find an expression for $e^*(k)$ which is given by

$$e^*(k) = e(k) + \frac{u_r(k) - u(k)}{b_{c_0}}. \tag{2.62}$$

Another form to mathematically express the previous Equation is:

$$e^*(k) = \begin{cases} f[u(k), u_r(k), e(k)] & \text{if } u_r(k) \neq u(k), \\ e(k) & \text{if } u_r(k) = u(k), \end{cases} \tag{2.63}$$

where $f[u(k), u_r(k), e(k)]$ is given by the right side of Equation (2.62). In this way, it is possible to find a block diagram representation of the strategy, as shown in Figure 12.

Although it is possible to represent the AW strategy by a block diagram, it is not possible to implement an equivalent structure due to the algebraic loop involving a nonlinearity.

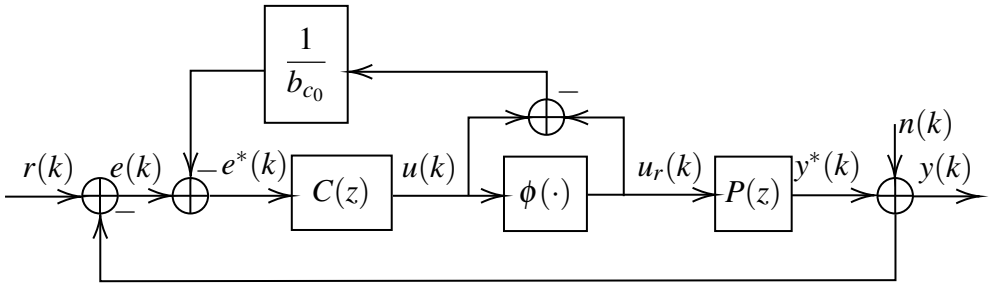


Figure 12 – Block diagram for the error recalculation AW (FLESCH, R.; NORMEY-RICO; FLESCH, C. 2017).

However, it is possible to work around this problem by isolating $e(k)$ in Equation (2.59) considering that $u(k) = u_r(k)$, resulting in (FLESCH, R.; NORMEY-RICO; FLESCH, C. 2017).

$$e(k) = \frac{u_r(k) - (b_{c_1}e(k-1) + \ldots + b_{c_q}e(k-q) - a_{c_1}u(k-1) - \ldots - a_{c_s}u(k-s))}{b_{c_0}}. \quad (2.64)$$

The error recalculation AW algorithm, considering a saturation of the control action magnitude, can be seen in Algorithm 2. The algorithm can also be extended to consider saturation in the increment of the control action.

Algorithm 2: Anti-windup: error recalculation

```
initialize variables;
repeat
    measure plant output y(k);
    e(k) ← r(k) − y(k);
    compute control action u(k);
    if u(k) ≤ u_min or u(k) ≥ u_max then
        if u(k) > u_max then
            u(k) ← u_max;
        else if u(k) < u_min then
            u(k) ← u_min;
        aux1 ← u(k) + a_c1 u(k−1) + ... a_cs u(k−s);
        aux2 ← b_c1 e(k−1) + ... b_cq e(k−q);
        e(k) ← (aux1 − aux2)/b_c0;
    apply u(k) to the plant;
    update variables;
    k ← k + 1;
until controller is stopped;
```

An important feature of this AW strategy is that it performs well in systems subject to measurement noise and operating near the saturation region of the process, as detailed in Silva, Flesch and Normey-Rico (2018). To demonstrate this property, the following example considers a discrete FOPDT model given by

$$P(z) = \frac{0.09}{z - 0.9}z^{-10}. \quad (2.65)$$

For this example, a discrete-time PID controller, $C_{PID}(z)$, presented in Equation (2.17) and tuned based on a low-frequency approximation of the FSP, as presented in Section 2.1.3, is used

$$C_{PID}(z) = \frac{2.96z^2 - 5.08z + 2.17}{z^2 - 1.39z + 0.39}. \tag{2.66}$$

The example compares the closed-loop performance of three AW strategies: incremental algorithm, back-calculation and error recalculation. The considered process is subject to constraints on the control signal amplitude, with lower and upper limits of $u_{\min} = 0$ and $u_{\max} = 1$, respectively, and also considering that the system has measurement noise with a normal distribution, signal-to-noise ratio (SNR) of 13 dB, and variance of 0.05. The tuning parameter, T_t, for the back-calculation strategy was chosen using the rule proposed by Åström and Hagglund (1995), resulting in $T_t = 0.68$ s.

The simulation of the system is shown in Figure 13, comparing the performance of the three strategies for a step reference signal with an amplitude of 0.9 applied at time $t = 12$ s.

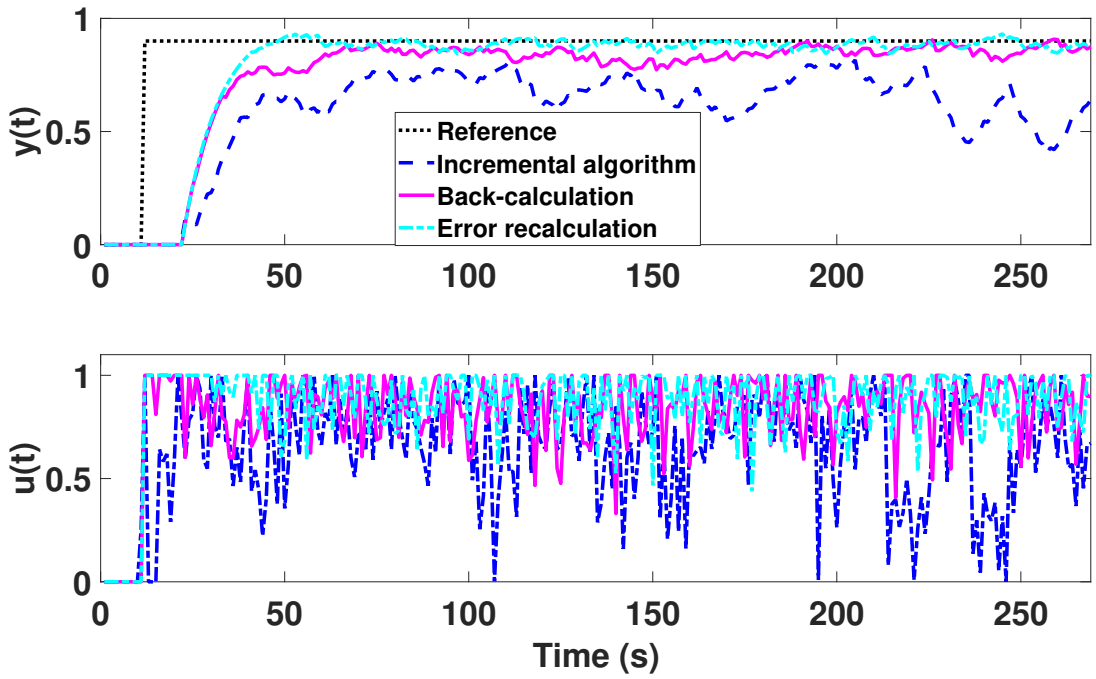


Figure 13 – Comparative analysis between the performance of the AW techniques considering noisy measurements.

As can be seen in Figure 13, the ER strategy performs considerably better compared to the other two, which exhibit steady-state error. The better behavior of the ER technique for the case which considers measurement noise can be explained based on Equation (2.67), which calculates the error signal $e(k)$ by considering the reference signal $r(k)$, the plant output $y^*(k)$, and the noise signal $n(k)$.

$$e(k) = r(k) - [y^*(k) + n(k)]. \tag{2.67}$$

Considering the case where there is a constraint on the maximum amplitude of the control signal $u_{\max}$, when the control signal is close to the saturation region, any noise signal

in the system can cause the control signal to saturate. This statement can be mathematically represented as

$$\begin{aligned} u(k) =& b_{c_0}\left[r(k) - y^*(k) - n(k)\right] + b_{c_1}e(k-1) + \ldots + b_{c_r}e(k-r) - \\ & a_{c_1}u(k-1) - a_{c_2}u(k-2) - \ldots - a_{c_s}u(k-s) > u_{\max}, \end{aligned} \tag{2.68}$$

which is obtained by substituting Equation (2.67) in Equation (2.59).

In ER strategy, if the condition presented in Equation (2.68) is satisfied, $e(k)$ is recalculated so that the new control signal respects the equality $u(k) = u_{\max}$, which is equivalent to

$$\begin{aligned} u(k) =& b_{c_0}\left[r(k) - y^*(k) - n^*(k)\right] + b_{c_1}e(k-1) + \ldots + b_{c_r}e(k-r) - \\ & a_{c_1}u(k-1) - a_{c_2}u(k-2) - \ldots - a_{c_s}u(k-s), \end{aligned} \tag{2.69}$$

with $n^*(k) < n(k)$. Thus, the equivalent noise of the system is reduced, making the control signal less influenced by the noise. This results in an improvement in the closed-loop system performance, making the process output very close to the reference signal. This feature is not observed in the other two AW techniques, as they are based on the current control signal modification and do not recalculate the current error signal. This causes inconsistency between past error signals and past control signals applied to the plant if the control signal saturates in the next sampling periods. In the case of processes with noisy measurements, this results in an increase in control signal variability, making the mean of this signal not large enough to bring the process output to its optimal operating point, which can be observed in Figure 13 for the incremental algorithm and back-calculation strategies. This undesirable effect can also be observed in other control strategies, such as MPC (CAMACHO; BORDONS, 2013).

2.3.4 Reference governor

The reference governor (RG) approach is a method for regulating the inputs of a control system to ensure that they meet specified constraints while tracking a desired reference signal. The approach involves the use of a second control loop, referred to as the "governor", similar to the one presented in ER method, that modifies the reference signal used by the primary control loop.

It is assumed that the RG operates at a higher hierarchical level, with the primary control system already designed to stabilize the plant and provide a desired tracking performance and disturbance attenuation in the absence of constraints. The constraint fulfillment task is then delegated to the RG, which modifies the reference supplied to the primary control system as necessary to enforce the fulfillment of the constraints.

The formulation of the RG method can be done based on Equation (2.59), but now considering $e(k) = r(k) - y(k)$ which results in

$$\begin{aligned} u(k) =& b_{c_0}(r(k) - y(k)) + b_{c_1}(r(k-1) - y(k-1)) + \ldots + b_{c_q}(r(k-q) - y(k-q)) \\ & -a_{c_1}u(k-1) - \ldots - a_{c_s}u(k-s), \end{aligned} \tag{2.70}$$

where $r(k)$ is the reference signal and $y(k)$ is the measured output of the system.

The analysis of the RG approach is similar to the one presented in Section 2.3.3 for the ER approach: if $u(k) \neq u_r(k)$, the current reference signal $r(k)$ must be recalculated to enforce $u(k) = u_r(k)$. Considering that $r(k)^*$ represents the expected value for the current reference signal so that the equality $u(k) = u_r(k)$ is respected, then $r^*(k)$ can be computed as

$$r^*(k) = r(k) + \frac{u_r(k) - u(k)}{b_{c_0}}, \tag{2.71}$$

or as

$$r^*(k) = \begin{cases} f[u(k),u_r(k),r(k)] & \text{if } u_r(k) \neq u(k), \\ r(k) & \text{if } u_r(k) = u(k), \end{cases} \tag{2.72}$$

with $f[u(k),u_r(k),r(k)]$ given by the right-side expression of Equation (2.71).

In this way, the block diagram for the resulting control system can be represented as shown in Figure 14.

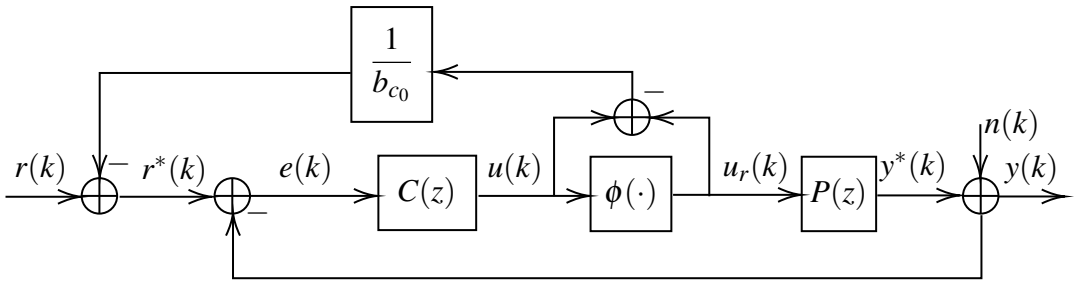


Figure 14 – Block diagram for the RG method

Similarly to the ER, the block diagram of RG method also presents an algebraic loop with a nonlinearity, so the practical implementation of the method can be done isolating $r(k)$ in Equation (2.70) and forcing that $u(k) = u_r(k)$ only if $u(k) \neq u_r(k)$, resulting in

$$r(k) = y(k) + \frac{u_r(k) - u_{\text{aux}}}{b_{c_0}}, \tag{2.73}$$

where

$$u_{\text{aux}} = b_{c_1}(r(k-1) - y(k-1)) + \ldots + b_{c_q}(r(k-q) - y(k-q)) - a_{c_1}u(k-1) - \ldots - a_{c_s}u(k-s). \tag{2.74}$$

The reference governor algorithm for magnitude constraints can be seen in Algorithm 3. It is important to note that this algorithm can also be extended to the case with constraints in the increment of control action.

2.4 PERFORMANCE AND ROBUSTNESS INDEXES

Performance and robustness analyses are both critical components in the design of control systems. Performance analysis is concerned with assessing the capability of the system

Algorithm 3: Reference governor method

initialize variables;
repeat
  measure plant output $y(k)$;
  $e(k) \leftarrow r(k) - y(k)$;
  compute control action $u(k)$;
  if $u(k) \leq u_{\min}$ **or** $u(k) \geq u_{\max}$ **then**
    if $u(k) > u_{\max}$ **then**
      $u(k) \leftarrow u_{\max}$;
    else if $u(k) < u_{\min}$ **then**
      $u(k) \leftarrow u_{\min}$;
    $u_{\text{aux}_1} \leftarrow b_{c_1}(r(k-1) - y(k-1)) + \ldots + b_{c_q}(r(k-q) - y(k-q))$;
    $u_{\text{aux}_2} \leftarrow -a_{c_1}u(k-1) - \ldots - a_{c_s}u(k-s)$;
    $r(k) \leftarrow y(k) + \frac{u(k) - (u_{\text{aux}_1} + u_{\text{aux}_2})}{b_{c_0}}$
  apply $u(k)$ to the plant;
  update variables;
  $k \leftarrow k+1$;
until *controller is stopped*;

to meet specified performance objectives, which could include objectives such as tracking a desired reference trajectory, rejecting external disturbances and achieving a desirable response speed. The goal of robustness analysis is to ensure that a control system is stable in the presence of uncertainty or disturbances.

In Sections 2.4.1 and 2.4.2, the performance and robustness metrics employed for the analysis of the controllers implemented in this study are presented.

2.4.1 Performance index

To make a quantitative analysis of the performance of some control strategies in this work, a cost function, J, which considers the integral of absolute error (IAE) of the reference response and the disturbance response, is used. This cost function is defined as

$$J = \frac{1}{2}\left[\int_{t_r+L_n}^{t_p} |r(t) - y(t)|\mathrm{d}t + \int_{t_p+2L_n}^{\infty} |r(t) - y(t)|\mathrm{d}t\right], \tag{2.75}$$

where t_r is the time at which the reference signal is applied, t_p is the time at which the disturbance signal is applied, $r(t)$ is the reference signal, $y(t)$ is the process output, and L_n is the dead time. It is important to note that the IAE index must be measured from the instant $t_r + L_n$ to t_p for the reference response and from the instant $t_p + 2L_n$ to infinity for the disturbance response, because no controller is able to modify the process output before $t_r + L_n$ for the case of change in reference and before $t_p + 2L_n$ for the case of disturbance rejection. Furthermore, it is important to ensure that t_p is sufficiently large to allow the process output to reach a steady state after a change in the reference signal.

2.4.2 Robustness indexes

For the robustness analysis of the controllers implemented in this work, the robustness index, $R_I(\omega)$, the delay margin, D_M, and the maximum sensitivity, M_s, are used. The $R_I(\omega)$ index is given by (ÅSTRÖM; HAGGLUND, 1995)

$$R_I(\omega) = \frac{|1 + C(j\omega)P_n(j\omega)|}{|C(j\omega)P_n(j\omega)|} \quad \forall\, \omega \geq 0. \tag{2.76}$$

The robustness index is used to verify if the robust stability condition $R_I(\omega) > \overline{\delta P}(\omega)$, for all $\omega \geq 0$, is satisfied, where $\overline{\delta P}(\omega)$ is the norm of the multiplicative uncertainty $P(\omega) = P_n(\omega)[1 + \delta P(\omega)]$, $\overline{\delta P}(\omega) \geq |\delta P(\omega)|$, for all $\omega \geq 0$, $C(s)$ is the controller, $P_n(s)$ is the process model used to tune the controller $C(s)$ (NORMEY-RICO; CAMACHO, 2007).

The index D_M is given by (PALMOR, 1980)

$$D_M = \frac{P_M}{\omega_c}, \tag{2.77}$$

where P_M is the phase margin, given in radians, and ω_c is the cut-off frequency, given in radians per unit time. This index is used to evaluate the system robustness against modeling errors in dead time (NORMEY-RICO; CAMACHO, 2007). The value of D_M represents the smallest delay value that can cause the process to become unstable (PALMOR, 1980).

The M_s index is generally used to define the robustness of a system when modeling errors are not estimated, and is given by (ÅSTRÖM; HAGGLUND, 1995)

$$M_s = \max_{\omega} |1 + C(j\omega)P_n(j\omega)|^{-1}, \tag{2.78}$$

where $C(j\omega)$ is the controller and $P_n(j\omega)$ is the process model used to tune $C(j\omega)$. According to Åström and Hagglund (1995), the typical values of M_s used for industrial applications are in the region between 1.2 to 2.0.

2.5 FINAL CONSIDERATIONS

In this chapter, some of the main strategies proposed in the literature for controlling processes with dead time, known in the literature as DTC, have been presented. The SP and one of its extensions, the FSP, have been highlighted as prominent techniques. Additionally, an approach for PID tuning based on a low-frequency approximation of the FSP has been discussed. The chapter has also explored the capabilities MPC strategy, specifically its ability to effectively handle dead time and constraints. Furthermore, the chapter has covered some techniques proposed in the literature for handling constrained processes, such as AW and governors. A discussion of AW strategies when used in systems with significant measurement noise has also been presented.

The following chapters of this work discuss the analysis and formulations addressed in three papers, covering topics related to the control of processes with dead time, constraints, modeling errors, and measurement noise.

3 PAPER 1 - CONTROLLING INDUSTRIAL DEAD-TIME SYSTEMS: WHEN TO USE A PID OR AN ADVANCED CONTROLLER

In this chapter the analysis and formulations of the paper "Controlling Industrial dead-time systems: when to use a PID or an advanced controller", which was published in the Journal of ISA Transactions, are presented. The discussion includes a comparative analysis between PID, DTC and MPC strategies, in terms of performance and robustness, in order to provide guidelines to choose which method is more conveninent to be used based on the characteristics of the process.

3.1 CONTROL OF DEAD-TIME PROCESSES

As shown in Section 2.1.1, the SP present good performance when used to control dead-time processes. For a system composed of a pure dead-time, given by

$$P(s) = e^{-Ls}, \tag{3.1}$$

and considering an ideal SP, i.e. with a primary controller, $C_{\mathrm{SP}}(s) = K$, with infinite gain K and with a perfect model of the plant ($P_n(s) = P(s)$), the SP is capable of providing the ideal solution for the system, which is a step-like response, after one dead time unit, L_n, for setpoint changes, and after two dead time units, for input (load) disturbances. These properties can be observed from the analysis of the closed-loop transfer functions for reference, $H_{yr}(s)$, and for input disturbance rejection, $H_{yq}(s)$, given respectively by

$$H_{yr}(s) = \frac{C_{\mathrm{SP}}(s)P_n(s)}{1 + C_{\mathrm{SP}}(s)G_n(s)} = \frac{K}{1+K}e^{-L_n s} \overset{K\to\infty}{\approx} e^{-L_n s}, \tag{3.2}$$

$$H_{yq}(s) = P_n(s)\left[1 - \frac{C_{\mathrm{SP}}(s)P_n(s)}{1 + C_{\mathrm{SP}}(s)G_n(s)}\right] \overset{K\to\infty}{\approx} e^{-L_n s}(1 - e^{-L_n s}). \tag{3.3}$$

It is important to note that, even for the ideal case, it is not possible to eliminate one delay from the setpoint response and two delays from the disturbance response. However, for systems with dynamics different from that of a pure delay, the SP performance for disturbance rejection is degraded, and the method is not capable of providing the ideal response, because the open-loop poles appear in the closed-loop response. For example, the transfer function from an input disturbance to the process output, $H_{yq}(s)$, of the ideal SP for a FOPDT process,

$$P(s) = \frac{1}{\tau s + 1}e^{-Ls}, \tag{3.4}$$

where τ is the time constant of the process, is given by

$$H_{yq}(s) = \frac{1}{\tau s + 1}e^{-L_n s}(1 - e^{-L_n s}). \tag{3.5}$$

In Equation (3.5) it is possible to note that the time constant of the process appears in the closed-loop dynamics. Furthermore, as presented in Section 2.1.1, the original SP structure

is not capable of controlling integrating and unstable processes. A simple solution for these drawbacks, discussed in Section 2.1.2, is known as FSP, which adds a robustness filter, $F_r(s)$, in the prediction error of the original SP structure, as shown in Figure 4.

Considering the nominal case (i.e. $P_n(s) = P(s)$), the closed-loop transfer functions for the reference and disturbance rejection of the FSP strategy are given by

$$H_{yr}(s) = \frac{C_{\text{FSP}}(s)P_n(s)}{1 + C_{\text{FSP}}(s)G_n(s)}, \tag{3.6}$$

$$H_{yq}(s) = P_n(s)\left[1 - \frac{C_{\text{FSP}}(s)P_n(s)F_r(s)}{1 + C_{\text{FSP}}(s)G_n(s)}\right]. \tag{3.7}$$

The tuning of $F_r(s)$ in the FSP allows the elimination the open-loop poles of $P_n(s)$ from $H_{yq}(s)$. Thus, if an ideal primary controller, $C_{\text{FSP}}(s)$, with infinite gain and an ideal filter, $F_r(s)$, are used, the ideal closed-loop response can be obtained.

In order to show the properties of SP and FSP, the following example considers an IPDT plant which is given by

$$P(s) = \frac{e^{-s}}{s}, \tag{3.8}$$

with time given in seconds. For this example, a primary controller $C_{\text{FSP}}(s) = K$ with infinite gain and an FSP robustness filter, $F_r(s) = 1 + s$, are used. With this tuning, the reference response, considering the nominal case, is given by

$$H_{yr}(s) = e^{-s}, \tag{3.9}$$

which is valid for both strategies. The disturbance responses for the SP and FSP, respectively, are given by

$$H_{yq_{\text{SP}}}(s) = \frac{e^{-s}}{s}(1 - e^{-s}), \tag{3.10}$$

$$H_{yq_{\text{FSP}}}(s) = \frac{e^{-s}}{s} - \frac{e^{-2s}}{s} - e^{-2s}. \tag{3.11}$$

Figure 15 shows the simulation of the system, comparing the performance of both techniques, for a step reference with amplitude of 1, applied at $t = 1\,\text{s}$, and for a step disturbance with amplitude of 0.5 applied at $t = 20\,\text{s}$.

As can be seen in Figure 15, the SP is not able to reject the step disturbance and the system ends up showing steady-state error. On the other hand, the FSP rejects the disturbance immediately after two dead time units (2 s). This is possible because the FSP allows tuning the robustness filter, $F_r(s)$, so that the equivalent controller presents integral action, which is not possible considering the original SP structure.

3.1.1 Comparative analysis between a PID and an FSP

This section presents a comparative analysis in terms of performance and robustness of a 2DOF PID controller and an FSP for unconstrained delay processes. The objective of this

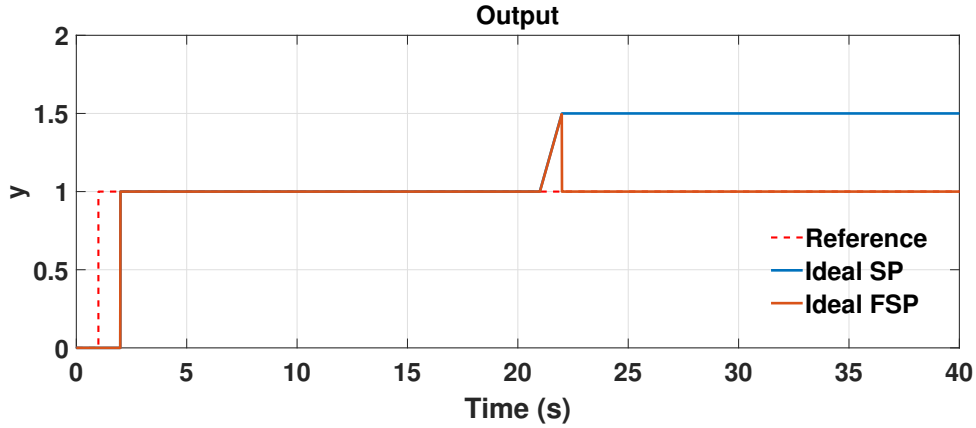


Figure 15 – Performance comparison between SP and FSP for an ideal case

analysis is to give some guidelines to the control designer in defining whether to use a PID or an FSP.

The tuning of PID and FSP controllers, discussed in this analysis, uses the same methodology presented in Section 2.1.3. As both tunings are based on the choice of the parameter T_0, it becomes easier to compare these strategies to more aggressive or more conservative tunings. The analysis considers the relationship between robustness, using the indices $R_I(\omega)$ and D_M, and performance, using the performance index J, as a function of the ratio T_0/L_n. Furthermore, for robustness analysis, the modeling error $\delta P(\omega)$ is also considered in the analysis.

For all cases $\delta P(\omega)$ is computed for a modeling error of 2% in the gain, time constant (when applicable) and dead time. In order to present a general analysis, a normalized model of the process with unitary gain and time constant (except for IPDT) is considered. The gain does not change the dynamics of the process, since the controller gain can be tuned accordingly to compensate for different gains of the process. The relation which really matters is between the fast model dynamics and the time delay, so it is possible to keep the dynamics fixed and change just the delay value without loss of generality. The three process models considered are: stable, $P_s(s) = \frac{e^{-L_n s}}{s+1}$, integrating, $P_i(s) = \frac{e^{-L_n s}}{s}$, and unstable, $P_u(s) = \frac{e^{-L_n s}}{s-1}$, with time given in seconds.

The first part of the comparative analysis considers a stable lag dominant (SLD)[1] process, with $L_n = 0.2\,\text{s}$, and a stable delay dominant (SDD)[2] process, with $L_n = 5\,\text{s}$. Figure 16 shows a comparative analysis between PID and FSP for both cases.

As can be seen in Figure 16, considering the SLD case, for a robust tuning of $T_0 = 0.3\,\text{s}$, the PID and FSP have similar robustness and performance properties. It is also possible to observe that when more robust the tuning, the greater is the similarity of the two techniques in terms of robustness and performance. In this case, the values of index D_M obtained for the controllers were $D_{M_{\text{PID}}} = 0.271\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.281\,\text{s}$, which confirms the equivalence of both strategies in terms of robustness for the same tuning. On the other hand, considering a tuning aiming fast responses, for example with $T_0 = 0.15\,\text{s}$, there is a significant improvement

[1] A process is lag dominant when $L_n \ll \tau$

[2] A process is delay dominant when $L_n \gg \tau$ (NORMEY-RICO; CAMACHO, 2007)

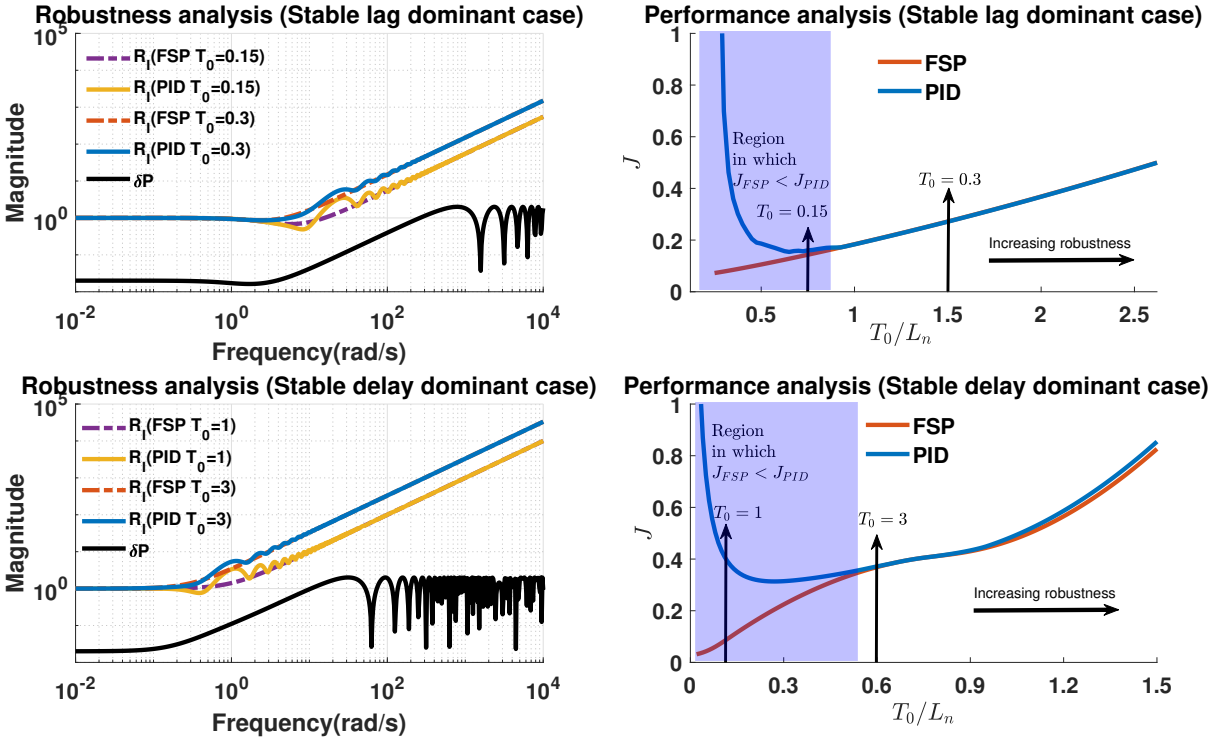


Figure 16 – Comparative analysis between a PID and FSP for the SDL and SDD cases

in performance when using FSP, as can be seen in the shaded area. Furthermore, a big difference in the robustness of the system can be observed for small values of T_0/L_n. The values of D_M obtained for the PID and the FSP were respectively $D_{M_{\text{PID}}} = 0.11\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.06\,\text{s}$, which shows that the FSP is more sensitive to uncertainty in relation to dead time.

The results in terms of robustness and performance, for the SDD case, when a robust tuning is considered ($T_0 = 3\,\text{s}$), are quite similar for both strategies. The values of D_M obtained for this tuning where very similar, resulting in $D_{M_{\text{PID}}} = 8.94\,\text{s}$ and $D_{M_{\text{FSP}}} = 9.23\,\text{s}$. As for the case that considers a tuning focused on performance ($T_0 = 1\,\text{s}$), again, the PID presented worse performance, however, better robustness in terms of dead time uncertainties when compared to FSP. The values of D_M obtained for both strategies were $D_{M_{\text{PID}}} = 5.52\,\text{s}$ and $D_{M_{\text{FSP}}} = 1.49\,\text{s}$.

The results in terms of robustness and performance, for the SDD case, when a robust tuning is considered ($T_0 = 3\,\text{s}$), are quite similar for both strategies. The values of D_M obtained for this tuning where very similar, resulting in $D_{M_{\text{PID}}} = 8.94\,\text{s}$ and $D_{M_{\text{FSP}}} = 9.23\,\text{s}$. As for the case that considers a tuning focused on performance ($T_0 = 1\,\text{s}$), again PID presented worse performance, however better robustness in terms of dead time uncertainties when compared to FSP. The values of D_M obtained for both strategies were $D_{M_{\text{PID}}} = 5.52\,\text{s}$ and $D_{M_{\text{FSP}}} = 1.49\,\text{s}$.

The second part of the comparative analysis considers a integrating case, $P_i(s)$, and an unstable case, $P_u(s)$, both with dead time of $L_n = 1\,\text{s}$. Figure 17 presents the performance and robustness analysis for both cases. As can be seen in Figure 17, the results of the integrating and unstable case were quite similar to those obtained in the SDL case. For the integrating case, the values of D_M obtained, considering a robust tuning ($T_0 = 1\,\text{s}$), were $D_{M_{\text{PID}}} = 0.53\,\text{s}$

and $D_{M_{\text{FSP}}} = 0.48\,\text{s}$. For the case that considers a tuning focusing on performance ($T_0 = 0.5s$), the values obtained were $D_{M_{\text{PID}}} = 0.192\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.142\,\text{s}$. For the unstable case, it is possible to observe that FSP presents a significant advantage in terms of both robustness and performance over the PID when robust and fast tuning solutions are considered. It is important to note that, along the whole interval of T_0/L_n analyzed, the performance of the FSP is better than the one of PID for the unstable case. The D_M values obtained for a robust tuning ($T_0 = 2\,\text{s}$) were $D_{M_{\text{PID}}} = 0.14\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.18\,\text{s}$, and for a fast tuning ($T_0 = 1.2\,\text{s}$) were $D_{M_{\text{PID}}} = 0.06\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.12\,\text{s}$.

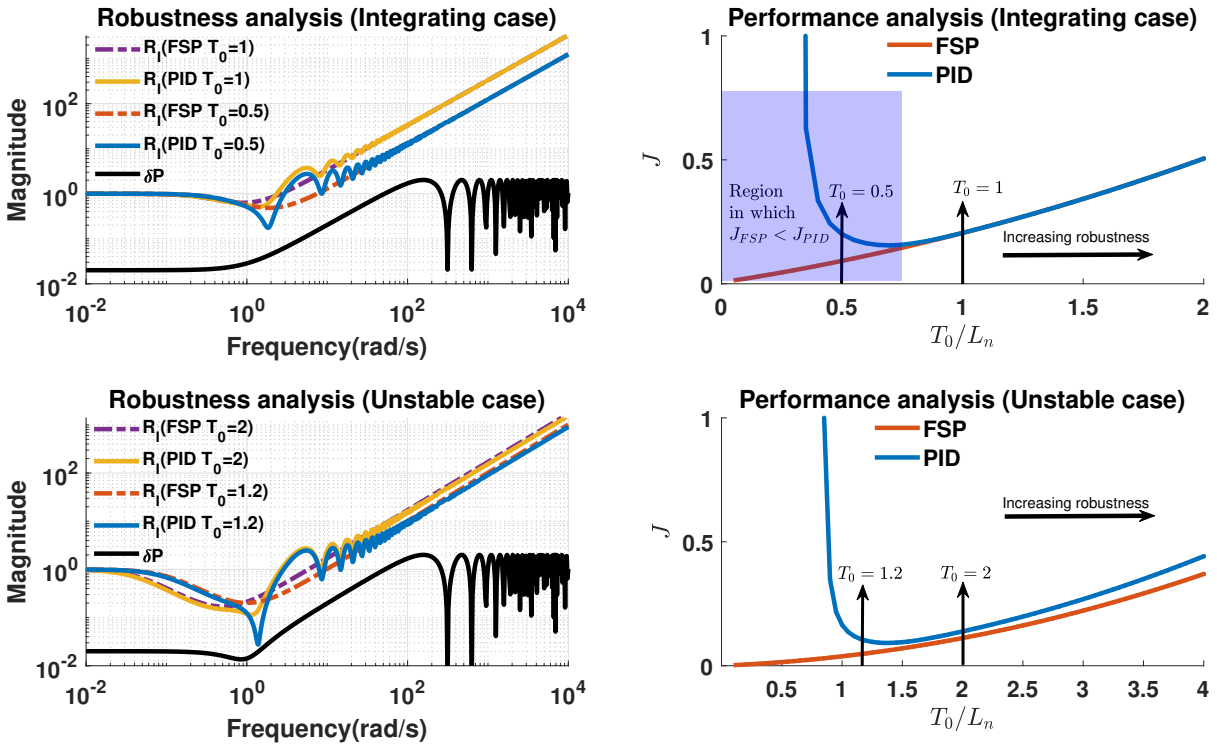


Figure 17 – Comparative analysis between PID and FSP for integrating and unstable cases

As can be seen in the previous examples, except for the unstable case, when robust solutions are considered, the performance and robustness properties of PID and FSP are practically the same. For solutions focused on fast response, the performance of the FSP is better than the one presented by PID even for processes with small delays (lag dominant). However, for these cases, the sensitivity to dead-time uncertainties for the FSP is significantly higher than the one presented by PID. This is due to an oscillatory behavior near the region of magnitude equal to 1 in the frequency response of the FSP loop transfer function, which causes a jump in the cutoff frequency ω_c to a higher value, resulting in a decrease in the D_M. The study of this phenomenon, known as *crossover proliferation*, can be seen in Horowitz (1983). A similar reduction in the D_M of SP was also observed in the works of Gudin and Mirkin (2007) and Skogestad (2018).

As a general conclusion of the results presented in this Section it can be stated that in most cases, when a high robustness system is necessary, a well-tuned PID is able of providing

similar results to an FSP, even if the process is delay dominant. On the other hand, when robustness is not a mandatory requirement and a fast tuning is required, the performance improvement of using an FSP instead of a PID is relevant and this result is valid even if the process is lag dominant. As this conclusion is valid for any constant ratio L_n/τ, it is possible to state that the advantages of using an FSP are more associated with the process modeling error than with the absolute value of dead time.

3.1.2 Case study

In order to validate the results presented in Section 3.1.1, a comparative analysis between a PID and an FSP for a particular example is presented. The example considers a second-order SDD process given by

$$P(s) = \frac{1}{s^2 + 0.8s + 1} e^{-2s}, \tag{3.12}$$

with time given in seconds, which was approximated by the following FOPDT model

$$P_m(s) = \frac{1}{0.62s + 1} e^{-2.7s}. \tag{3.13}$$

The example also considers measurement noise with normal distribution and variance of 0.02. The first part of the case study compares a PID and an FSP, in terms of performance. The tuning rule used for both controllers is the same as discussed in Section 2.1.3, in which both controllers are tuned using the same tuning parameter T_0. Considering a robust tuning, with $T_0 = 2\,\text{s}$, the PID parameters obtained were $k_c = 0.355$, $T_i = 1.758\,\text{s}$, $T_d = 1.35\,\text{s}$, $\alpha = 1{,}305$ and a reference filter given by

$$F(s) = \frac{1.14(s + 0.5)}{s + 0.57}. \tag{3.14}$$

For the FSP, the primary controller $C_{\text{FSP}}(s)$ obtained was a PI with $k_c = 0.31$ and $T_i = 0.62\,\text{s}$ and the robustness filter used was $F_r(s) = 1/F(s)$.

Figure 18 shows the simulation of the system and compares the performance of both strategies for a unit-step reference signal applied at $t = 1\,\text{s}$ and a step disturbance with an amplitude of -0.2 applied at $t = 50\,\text{s}$. As can be seen in Figure 18, the performance of the two strategies is quite similar for both reference tracking and disturbance rejection. This similarity can also be observed by analyzing the equivalent performance index obtained for both strategies, whose values obtained were respectively $J_{\text{PID}} = 21.1$ and $J_{FSP} = 20.2$. In terms of robustness, the delay margin obtained for the PID and FSP, respectively, were $D_{M_{\text{PID}}} = 5.53\,\text{s}$ and $D_{M_{\text{FSP}}} = 5.66\,\text{s}$.

In order to obtain a faster performance, a new tuning of the controllers was considered. In this case, the tuning of the controllers was performed based on the perfect model of the process, i.e. $P_m(s) = P(s)$. The PID was tuned using SWORD, which is a tool used to perform robust tuning for PID controllers (GARPINGER; HÄGGLUND, 2015). Both controllers were tuned to get a value of $M_s = 2$. The parameters obtained for the PID were $k_c = 0.282$, $T_i =$

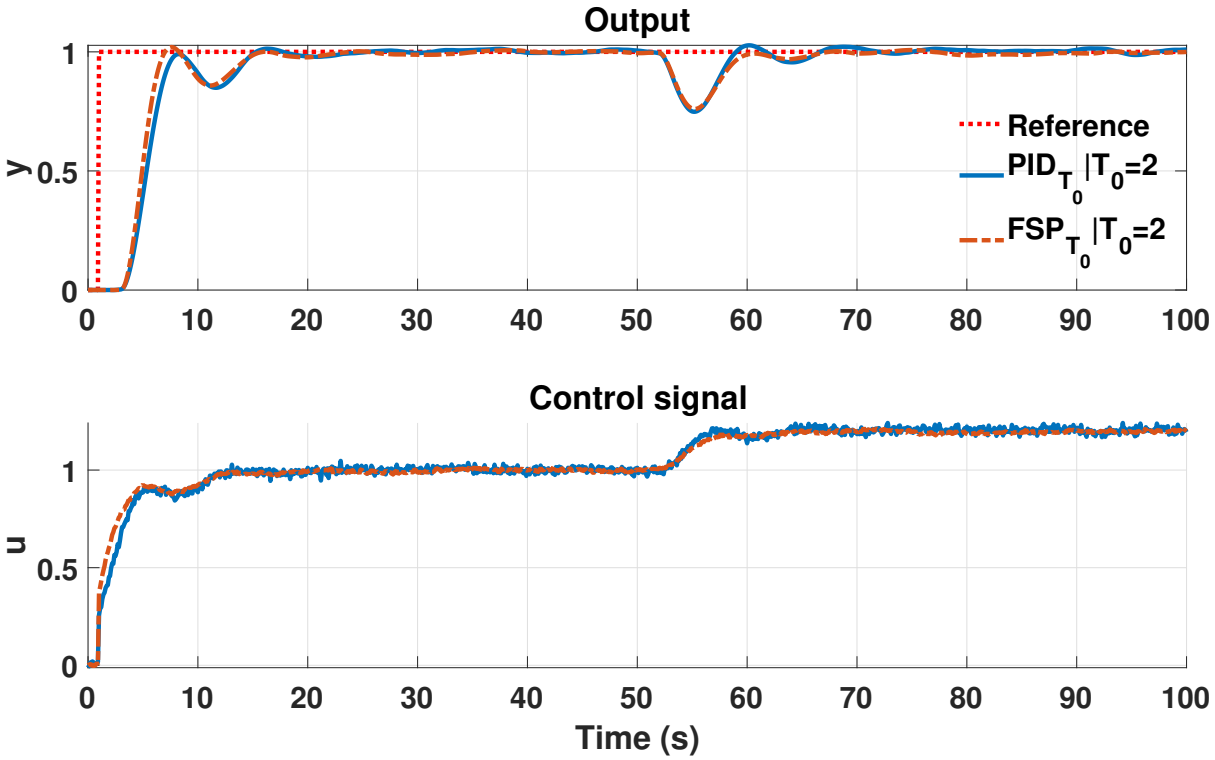


Figure 18 – Comparative analysis between PID and FSP for the unconstrained case (tuning for robustness)

0.992 s, $T_d = 1.984$ s with a second-order low-pass filter, $F_{PB}(s)$, given by

$$F_{PB}(s) = \frac{1}{0.035s^2 + 0.25s + 1}. \tag{3.15}$$

As the SWORD tool aims to optimize the disturbance rejection, the reference response obtained is quite oscillatory. In order to reduce this oscillation in transient response, a second-order reference filter, $F(s)$, was used and is given by

$$F(s) = \frac{s^2 + 1.4s + 1}{(s+1)^2}. \tag{3.16}$$

For the FSP, the primary controller, $C_{\text{FSP}}(s)$, and the robustness filter, $F_r(s)$, obtained considering an $M_s = 2$, were respectively

$$C_{\text{FSP}}(s) = \frac{75.13(s^2 + 2.03s + 1.96)}{s(s + 26.96)}, \tag{3.17}$$

$$F_r(s) = \frac{0.2(s + 11.16)}{s + 2.23}. \tag{3.18}$$

Figure 19 shows the simulation of the system considering the new tuning. As can be seen, the PID, when considering a tuning to accelerate the response, presents a very oscillatory behavior both for the reference tracking and disturbance rejection. On the other hand, the FSP with a tuning based on the perfect model of the process presents a very significant performance improvement in relation to PID, with lower overshoot and less oscillations in the disturbance

response. However, it is important to note that the FSP is considerably more sensitive to noise measurements, due to the high gain of the controller at high frequencies. The performance indices obtained for this case were $J_{\text{PID}} = 18.7$ and $J_{\text{FSP}} = 11.8$. The delay margin values obtained were $D_{M_{\text{PID}}} = 3.87\,\text{s}$ and $D_{M_{\text{FSP}}} = 0.52\,\text{s}$, which shows that the FSP is significantly less robust than the PID in terms of dead-time modeling uncertainties.

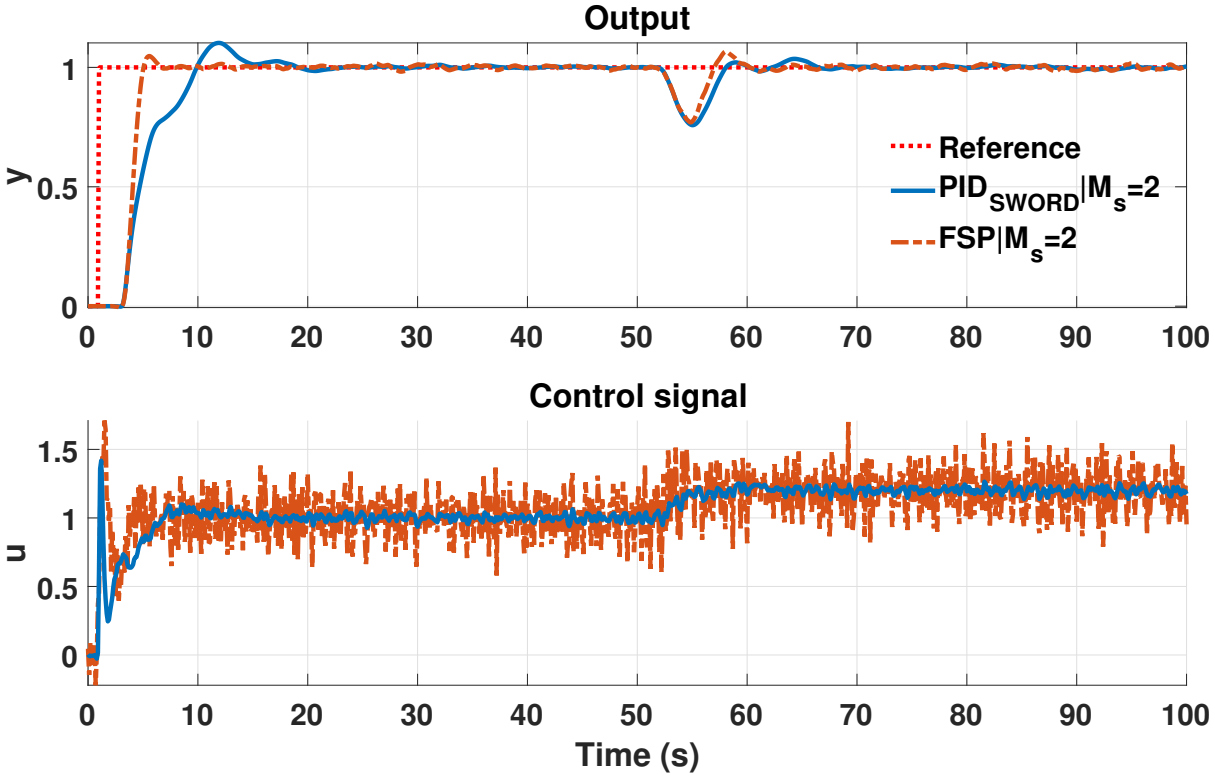


Figure 19 – Comparative analysis between PID and FSP for the constrained case (tuning for fast performance)

The results presented in this case study demonstrate that, considering scenarios where only a low-order model is available, which inadequately captures the process dynamics, and a robust solution must be employed, the performance improvement when using an FSP instead of a PID is not significant. However, when considering a well-represented system model that accurately characterizes the process dynamics, the use of the FSP can provide much better results than the PID, as shown in Figure 19. The presented results provide important information for the control designer in order to choose between a PID strategy or a more complex structure, such as an FSP, taking into account robustness specifications. However, in some cases, this analysis is not sufficient, due to the fact that real processes present physical limitations. As a result, Section 3.2.1 presents a similar analysis considering constrained processes.

3.2 CONTROL OF PROCESSES WITH DEAD TIME AND CONTRAINTS

This section compares PID with an AW scheme and GPC strategies in terms of performance and robustness, where constraints in the magnitude and rate of change of the control action and in the output of the process are considered. Firstly, a formulation that allows the PID

with AW to deal with several types of constraints is proposed. Then, a comparative analysis between PID with the proposed formulation and GPC for a process subjected to constraints is presented. For the sake of brevity, the simulation case study presented in this section just considers an IPDT process, but it is also valid for the FOPDT and UFOPDT cases.

3.2.1 Approach for control of process with constraints

As presented in Section 2.2.5, GPC takes into account process constraints in the formulation of the cost function to be minimized, which results in a QP problem that must be solved at each sampling interval to find the value of optimal control signal. The size of the control horizon, N_u, directly affects the complexity of the QP problem and consequently the computational effort to compute the solution. The literature presents some works that show that, typically, the choice of $N_u = 1$ is enough to control simple systems as stable and non-minimum phase processes (CLARKE; MOHTADI; TUFFS, 1987; KEYSER; IONESCU, 2003; CASTANO et al., 2015). In these cases, the resulting optimization process is much simpler to solve.

In case of PID controllers, when the process presents constraints, typically the information of these limitations is used *a posteriori* with the use of AW techniques in order to reduce the undesired effects caused by the presence of constraints. Based on the behavior of the GPC with a control horizon $N_u = 1$, an algorithm for PID controllers is presented below, which allows the calculation of the control signal for processes subject to constraints on the amplitude and rate of change of the control signal, and also in the process output. Considers a FOPDT model given by

$$(1 + az^{-1})y(k) = z^{-d}bu(k-1), \tag{3.19}$$

subjected to the same constraints as in (2.40). The main idea of this approach for computing a feasible control signal is to rewrite the constraints in terms of the control signal, $u(k)$. This can be done using the prediction model of the GPC strategy, presented in Equation (2.27), considering that $\Delta u(k) = u(k) - u(k-1)$ and, in addition, considering a control horizon $N_u = 1$. Thus, the constraints presented in (2.40) can be rewritten in the given form

$$\begin{aligned} u_{\min} \leq & u(k) \leq u_{\max}, && \forall k, \\ \Delta u_{\min} + u(k-1) \leq & u(k) \leq \Delta u_{\max} + u(k-1), && \forall k, \\ u_{y_{\min}} \leq & u(k) \leq u_{y_{\max}}, && \forall k, \end{aligned} \tag{3.20}$$

where $u_{y_{min}}$ and $u_{y_{max}}$ are the minimum and maximum control signals to maintain the predictions of the future outputs inside the constraint boundaries and must be determined at each sampling instant. To obtain these values, firstly, the future output predictions need to be computed by using Equation (2.27) with $I(z^{-1}) = 1$ and considering that the future noise signals are null. In order to compute the predictions from the instants $k+d+1$ up to $k+d+N$, where N is the prediction horizon, the values of the future control signal increments are needed. However, considering a control horizon $N_u = 1$, it is possible to state that

$$\Delta u(k+j) = 0, \text{ for } j \geq 1, \tag{3.21}$$

and, in this way, it is possible to compute the future output predictions as function of past process outputs and increments of past and current control signals. Therefore, the output predictions after the dead time can be obtained as

$$\begin{aligned}\hat{y}(k+d+j|k) = &(1-a)\hat{y}(k+d+j-1|k) + a\hat{y}(k+d+j-2|k) \\ &+ b\Delta u(k+j-1),\end{aligned} \tag{3.22}$$

for $j = 1 \ldots N$ and $\Delta u(k+j-1) = 0$ for $j > 1$. Another important aspect that must be taken into account is that, for dead-time processes, the current control action affects the output only after $d+1$ sampling instants have passed. For this reason, the predictions of future outputs used to compute $u_{y_{\min}}$ and $u_{y_{\max}}$ must be from instants $k+d+1$ up to $k+d+N$. Equation (3.22) can be rewritten as a function of $\hat{y}(k+d|k)$, $\hat{y}(k+d-1|k)$ and $\Delta u(k)$ as

$$\begin{aligned}\hat{y}(k+d+j|k) = &s_{j1}\hat{y}(k+d|k) + s_{j2}\hat{y}(k+d-1|k) \\ &+ g_j \Delta u(k),\end{aligned} \tag{3.23}$$

where

$$\begin{aligned}&s_{11} = 1-a, \\ &s_{12} = a, \\ &s_{ji} = \sum_{k=1}^{2} s_{1k} s_{(j-k)i}, \text{ with } s_{j0} = s_{0i} = 1, \\ &g_j = (a^{j-1} + a^{j-2} + \ldots + 1)b, \\ &\text{for } j = 1 \ldots N \text{ and } i = 1 \ldots 2.\end{aligned} \tag{3.24}$$

To compute $u_{y_{min}}$ and $u_{y_{max}}$, it is necessary to guarantee that the future output predictions satisfy

$$y_{\min} \leq \hat{y}(k+d+j|k) \leq y_{\max}. \tag{3.25}$$

By substituting Equation (3.23) in (3.25) and considering that $\Delta u(k) = u(k) - u(k-1)$, it is possible to obtain the minimum and maximum control signal values to be applied at time instant k to satisfy the output constraints at time instant $k+d+j$, $u_{y_{min}}(j)$ and $u_{y_{max}}(j)$ respectively, as

$$\begin{aligned}&u_{y_{\min}}(j) \geq \frac{y_{\min} - s_{j1}\hat{y}(k+d|k) - s_{j2}\hat{y}(k+d-1|k)}{g_j} + u(k-1), \\ &u_{y_{\max}}(j) \leq \frac{y_{\max} - s_{j1}\hat{y}(k+d|k) - s_{j2}\hat{y}(k+d-1|k)}{g_j} + u(k-1), \\ &\text{for } j = 1 \ldots N.\end{aligned} \tag{3.26}$$

From (3.26) it is possible to obtain the values for $u_{y_{\min}}$ and $u_{y_{\max}}$ as

$$\begin{aligned}&u_{y_{\min}} = \max\left\{u_{y_{\min}}(j)\right\}, \\ &u_{y_{\max}} = \min\left\{u_{y_{\max}}(j)\right\}, \\ &\text{for } j = 1 \ldots N,\end{aligned} \tag{3.27}$$

since a value in the feasible region of (3.27) guarantees that all the constraints defined in (3.26) are satisfied.

Finally, after computing the limits of the control signals for each of the constraints, it is possible to reformulate the conditions defined in (3.20) as

$$U_{\min} \leq u(k) \leq U_{\max}, \tag{3.28}$$

where

$$\begin{aligned} U_{\min} &= \max\{u_{\min}, \Delta u_{\min} + u(k-1), u_{y_{\min}}\}, \\ U_{\max} &= \min\{u_{\max}, \Delta u_{\max} + u(k-1), u_{y_{\max}}\} \end{aligned} \tag{3.29}$$

are the limits that define the feasible region for all constraints specified in (3.20). Furthermore, it is possible to use AW strategies in cases where the computed control signal is outside the region defined by $U_{\min}$ and $U_{\max}$.

In order to present a systematic way for computing the control signal for processes subjected to the constraints specified in Equation (3.20), the Algorithm 4 is presented below. It is important to note that this methodology can also be used for high-order processes, simply by modifying the coefficients used in Equation (3.23), based on the model used.

Algorithm 4: Constraints mapping

```
initialize variables;
repeat
    compute control action u(k);
    for j = 1 to d + N do
        compute predictions ŷ(k + j)
    end
    compute u_ymin e u_ymax using (3.27);
    compute U_min e U_max using (3.29);
    if u(k) ≤ U_min or u(k) ≥ U_max then
        use AW to recalculate u(k) ;
    end
    apply u(k) to the plant;
    update variables;
    k ← k + 1;
until until controller is stopped;
```

3.2.2 Comparative analysis between PID and GPC

This section presents a comparative analysis between a PID, using the constraint handling approach described in Section 3.2.1, and a GPC. The case study considers robustness and performance characteristics and aims to show that the analysis performed in Section 3.1.1 can also be extended to processes subjected to constraints.

For the comparative analysis, an IPDT process $P_i(s) = \frac{e^{-L_n s}}{s}$, with $L_n = 1$ is considered. In the first part of the analysis, the perfect model of the plant is used to tune the controllers, allowing a tuning with a high value of M_s index. The discrete-time representation of the system,

using a ZOH and a sampling period of 0.1 s, is given by

$$P_i(z) = \frac{0.1z^{-10}}{z-1}. \tag{3.30}$$

Both controllers were tuned to present fast responses, resulting in a value of $M_s = 3.8$. For this tuning, the parameters obtained for the PID were $k_c = 0.99$, $T_i = 2.44\,\text{s}$, $T_d = 0.5\,\text{s}$, $\alpha = 0.21$, $T_0 = 0.73\,\text{s}$ and a reference filter given by

$$F(s) = \frac{0.05(s+10)}{s+0.47}. \tag{3.31}$$

For the GPC, the tuning parameters used were $N = 9$, $N_u = 1$ e $\lambda = 5$.

The case study also considers that the system is subjected to constraints on magnitude and rate of change of the control signal, with $u_{\min} = -0.1$, $u_{\max} = +0.6$, $\Delta u_{\min} = -0.1$, $\Delta u_{\max} = +0.1$, and also constraints in the process output, with $y_{\min} = 0$ and $y_{\max} = 1.1$. To deal with the constraints, a PID is used with the method presented in Section 3.2.1, with a prediction horizon of $N = 9$ and with the error recalculation (ER) AW strategy, whose formulation is presented in Section 2.3.3. The choice of this strategy is due to its ease of implementation and its good performance in processes subjected to measurement noise (SILVA; FLESCH; NORMEY-RICO, 2018).

Figure 20 shows the simulation of the system considering a unit step reference signal applied to $t = 1\,\text{s}$, and a step load disturbance of amplitude -0.2 applied to $t = 50\,\text{s}$. As can

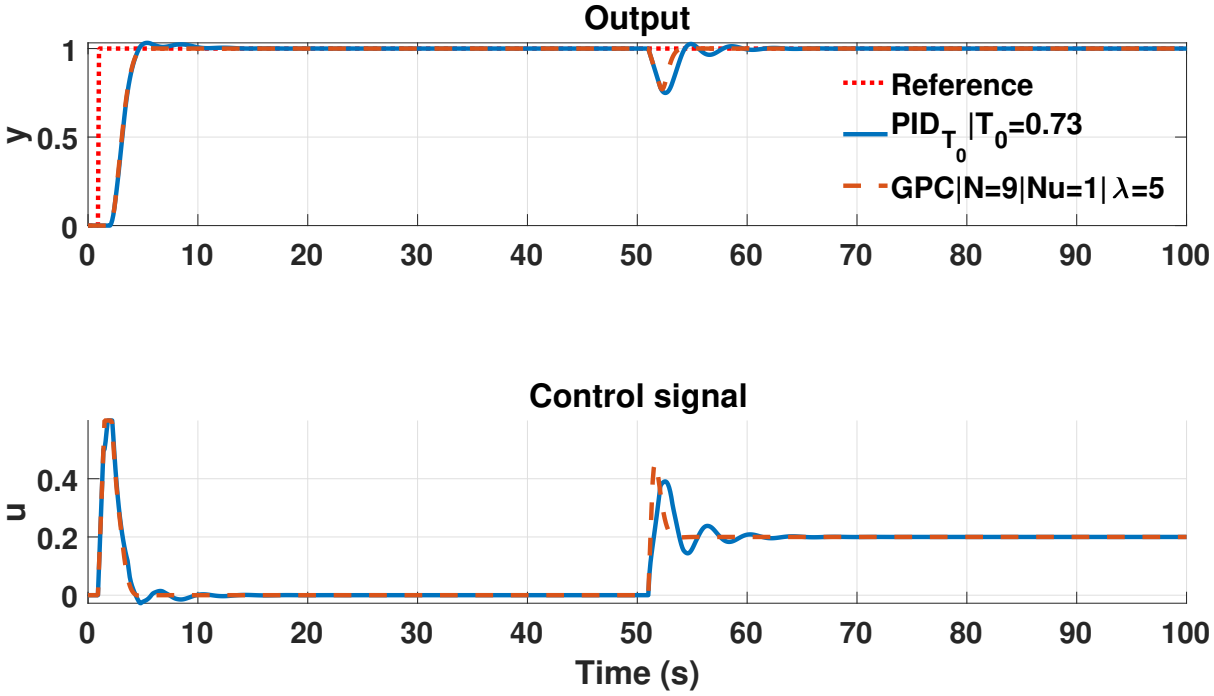


Figure 20 – Comparative analysis between PID and GPC for the case with no modeling errors

be seen, both controllers are able to handle the process constraints, however, the GPC performs better, presenting a reference response without overshoot and a disturbance rejection without

oscillations. The performance index obtained for the controllers were $J_{\text{PID}} = 9.8$ and $J_{\text{GPC}} = 7.7$. On the other hand, in terms of robustness, the GPC presented a very low value for the delay margin, $D_{M_{\text{GPC}}} = 0.09\,\text{s}$, when compared to the PID, which presented $D_{M_{\text{PID}}} = 0.37\,\text{s}$. Therefore, the price to pay for better performance in the GPC strategy is a less robust closed-loop system. It is important to point out that, for a given value of M_s, even using a larger value of the control horizon, N_u, the performance of the GPC cannot be improved.

The second part of the analysis aims to evaluate a tuning closer to those typically used in industrial environments, taking into account a modeling error of 20% in the system gain and dead time, and also a measurement noise with normal distribution and variance of 0.02. As GPC is quite sensitive to measurement noise, this case also compares the performance and robustness of the DTC-GPC strategy, which was discussed in Section 2.2.6. The controllers were tuned to obtain a robustness index of $M_s = 1.9$, which is within the range of values that are commonly used in industrial applications (ÅSTRÖM; HÄGGLUND, 2001). The PID parameters used for this tuning were $k_c = 0.63$, $T_i = 4.56\,\text{s}$, $T_d = 0.5\,\text{s}$, $\alpha = 0.44$, $T_0 = 1.78\,\text{s}$ and a reference filter, $F(s)$ given by

$$F(s) = \frac{0.39(s+0.56)}{s+0.22}. \tag{3.32}$$

For the GPC, the tuning parameters used were $N = 30$, $N_u = 1$ and $\lambda = 1250$. It is important to note that, for this case, using a low value of λ results in poor performance on the GPC, caused by the aggressive control action. For the DTC-GPC, the tuning used was $N = 15$, $N_u = 1$, $\lambda = 1$ and with a second-order robustness filter, $F_r(z)$, given by

$$F_r(z) = \frac{0.149(z-0.977)}{(z-0.944)^2}. \tag{3.33}$$

The DTC-GPC robustness filter allows tuning the controller with a lower value of λ than the one used in the original GPC.

As the controllers were tuned to obtain a robust solution, the control signal behaves more smoothly and does not exceed the saturation values considered in the analysis without modeling errors. In order to make the control signal saturate, the limits of the constraints on the amplitude and rate of change of the control signal were changed to $u_{\min} = -0.1$, $u_{\max} = +0.3$, $\Delta u_{\min} = -0.02$ and $\Delta u_{\max} = +0.02$. Furthermore, as in this case modeling errors are considered, the output constraints were implemented as soft constraints to avoid infeasibility in the GPC and DTC-GPC optimization process. To get a similar response on the PID, the output constraints were also relaxed for this case.

Figure 21 presents the result of the closed-loop simulation considering the new tuning of the controllers. As can be seen, the PID and the DTC-GPC obtained very similar performances, while the GPC presented a very oscillatory performance due to the measurement noise. The performance indices obtained for the controllers were $J_{\text{PID}} = 24.1$, $J_{\text{GPC}} = 66.5$ and $J_{\text{DTC-GPC}} = 24.4$. In terms of robustness, the three controllers obtained similar delay margin values, with $D_{M_{\text{PID}}} = 1.09$, $D_{M_{\text{GPC}}} = 0.93$, and $D_{M_{\text{DTC-GPC}}} = 1.08$.

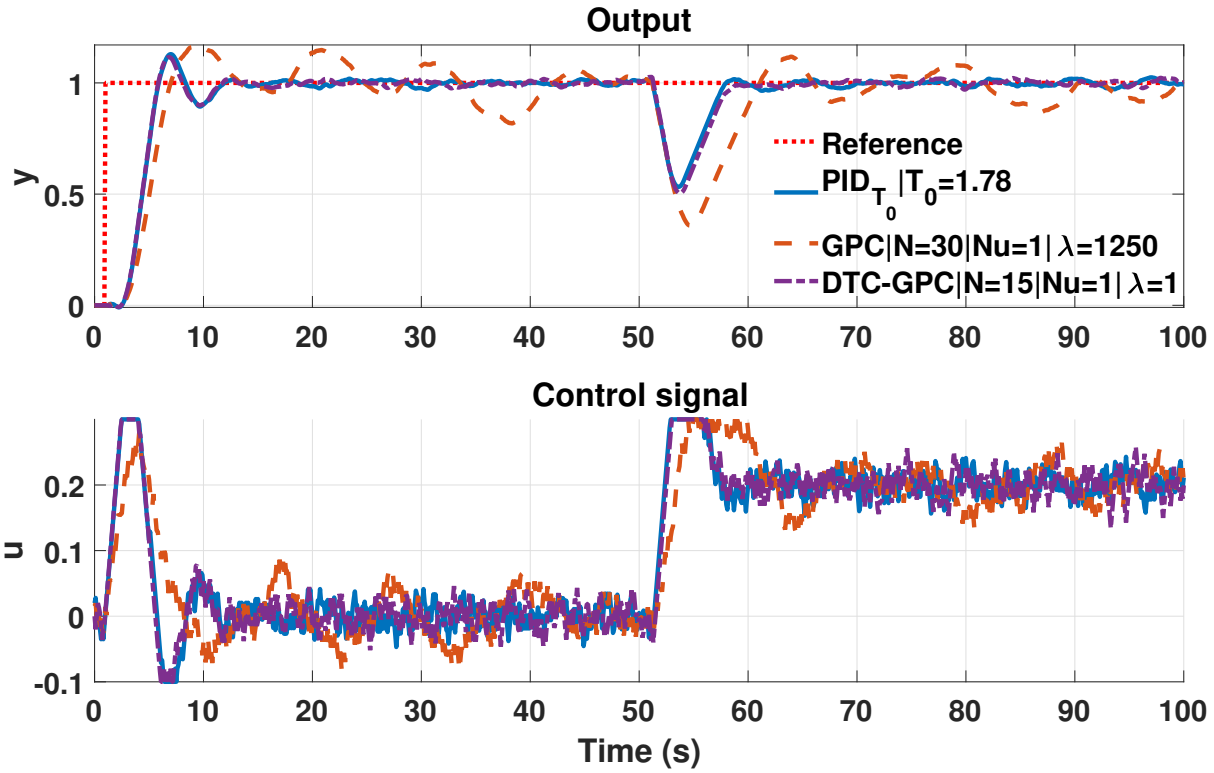


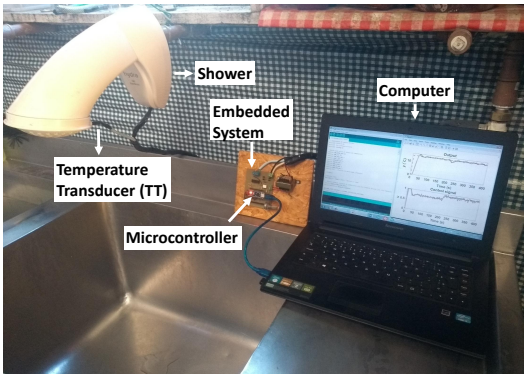
Figure 21 – Comparative analysis between PID and GPC for the case with modeling error and measurement noise

The results of this case study reinforce the central idea that, in situations where the available model inadequately captures the dynamics of the plant, which makes it necessary to use robust solutions, the PID strategy is the best choice, due to its simplicity and good relationship between performance and robustness, even in cases where the process is subjected to constraints.

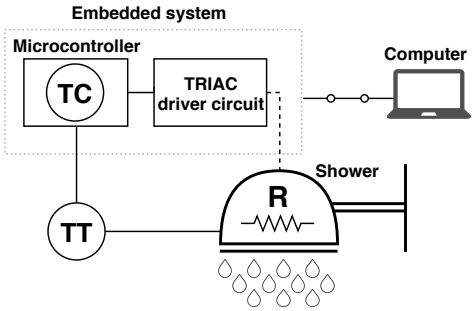
3.2.3 Experimental case study

This section presents an experimental case study which compares the performance of a PID with error recalculation AW and a GPC for a real process subjected to control signal amplitude constraints. The process considered is composed of an electronic shower and an embedded system, where the process variable is the water temperature variation and the controlled variable is the number of trigger pulses, in a period of 1 s, applied to a TRIAC (triode for alternating current) to control the power supplied for the power resistor of the shower. A picture of the process and a general diagram are shown in Figure 22, where TT is the temperature transducer, TC is the temperature controller and R is the shower heating resistor. The embedded system contains a microcontroller ATmega328P and the TRIAC driver circuit, which is composed of a zero crossing detector, an optoisolator and passive electrical components.

The embedded system is responsible for reading the water temperature using a temperature transducer and applying the control action to the TRIAC. The PID algorithm was implemented directly in the embedded system, while the GPC method was implemented in



Picture of the process



General process diagram

Figure 22 – Illustration of the experiment

a computer which uses an optimization algorithm to deal with process constraints. For both controllers the sampling period used is the same.

3.2.3.1 Model identification

The control action $u(t)$ was normalized, so that $u(t) = 1$ represents the maximum number of triggers (120 electrical network half-cycles), that is, the maximum power applied to the power resistor, and $u(t) = 0$ represents that no trigger should be performed, that is, the minimum power. The process output represents the temperature variation, in Celsius degrees (°C), in relation to the initial temperature of the water. The model was identified using power steps with different amplitudes, resulting in a FOPDT model given by

$$\frac{Y(s)}{U(s)} = \frac{18.7e^{-8s}}{13.7s+1}, \tag{3.34}$$

with time given in seconds.

Figure 23 shows that the approximated model represents adequately the dynamics of the system. It is important to note that the values of $y(^\circ\text{C})$ in Figure 23 are relative to the initial temperature of the water.

In order to obtain the discrete-time model of the plant, a ZOH and a sampling period $T_s = 1\,\text{s}$ were used, resulting in

$$\frac{Y(z)}{U(z)} = \frac{1.316z^{-8}}{z-0.923}. \tag{3.35}$$

The constraints taken into account in this process are the magnitude of the control signal, where $u_{\min} = 0$ and $u_{\max} = 1$.

3.2.3.2 Experimental results

Based on the identified process model, the PID and GPC controllers were tuned to obtain responses for reference tracking and disturbance rejection without oscillations. The PID tuning parameters used are $k_c = 0.06$, $T_i = 12.41\,\text{s}$, $T_d = 4\,\text{s}$, $\alpha = 0.41$, with $T_0 = 8\,\text{s}$. The PID

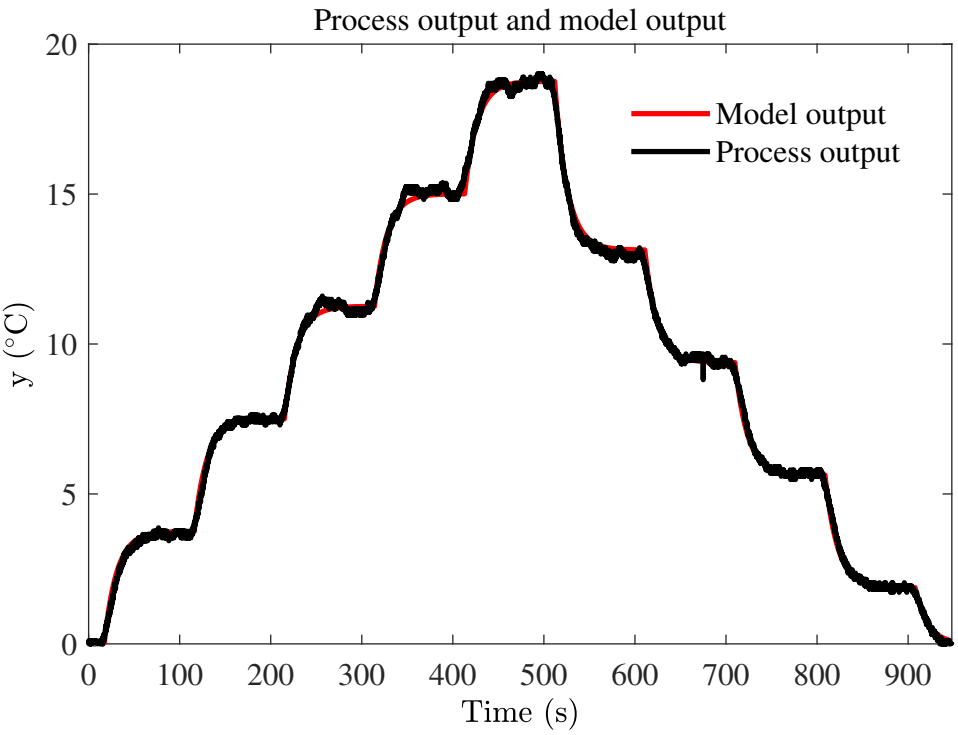


Figure 23 – Comparison between the model output and process output

uses the constraint handling approach proposed in Section 3.2.1 and the ER AW strategy to deal with the system constraints imposed by the actuator. The GPC tuning parameters used were $N = 60$, $N_u = 10$, with the weighting of the control increment $\lambda = 2100$. With this tuning, the indices M_s and D_M obtained for the PID and the GPC were $M_{s_{\text{PID}}} = 1.8$, $D_{M_{\text{PID}}} = 12\,\text{s}$, $M_{s_{\text{GPC}}} = 2.1$, and $D_{M_{\text{GPC}}} = 5\,\text{s}$. In addition, a new GPC tuning, aiming to speed up the response in relation to the first one, with $\lambda = 700$, is considered. The robustness indices obtained for this tuning are $M_{s_{\text{GPC}}} = 2.54$ and $D_{M_{\text{GPC}}} = 2.15\,\text{s}$.

Figure 24 shows the experimental results for a reference step with amplitude of $14\,^{\circ}\text{C}$ at $t = 5\,\text{s}$, with respect to the initial temperature, and a load disturbance step with amplitude -0.15 at $t = 200\,\text{s}$. As shown, the performance of the PID and GPC with the first tuning are similar both for reference and disturbance responses ($J_{\text{PID}} = 108.2$ and $J_{\text{GPC}} = 114.9$). Although both responses can be considered equivalent, PID presents a control signal with smaller variability, which results in a better performance index value. It is possible to improve the response presented by GPC by using a more aggressive tuning ($J_{\text{GPC}} = 107.3$), but the performance improvement is small when compared to both the GPC with the first tuning and the PID. Furthermore, the measurement noise affects more the control signal for the second tuning.

As can be seen, the PID and GPC responses for the first tune are quite similar in both reference tracking and disturbance rejection. It is also important to note that the variability of the control action in the PID is smaller than the one shown by GPC, because, considering the tuning used, the PID has a better ability to reject measurement noise. The performance indices obtained for this tuning were $J_{\text{PID}} = 108.2$ and $J_{\text{GPC}} = 114.9$. It is also important to note that

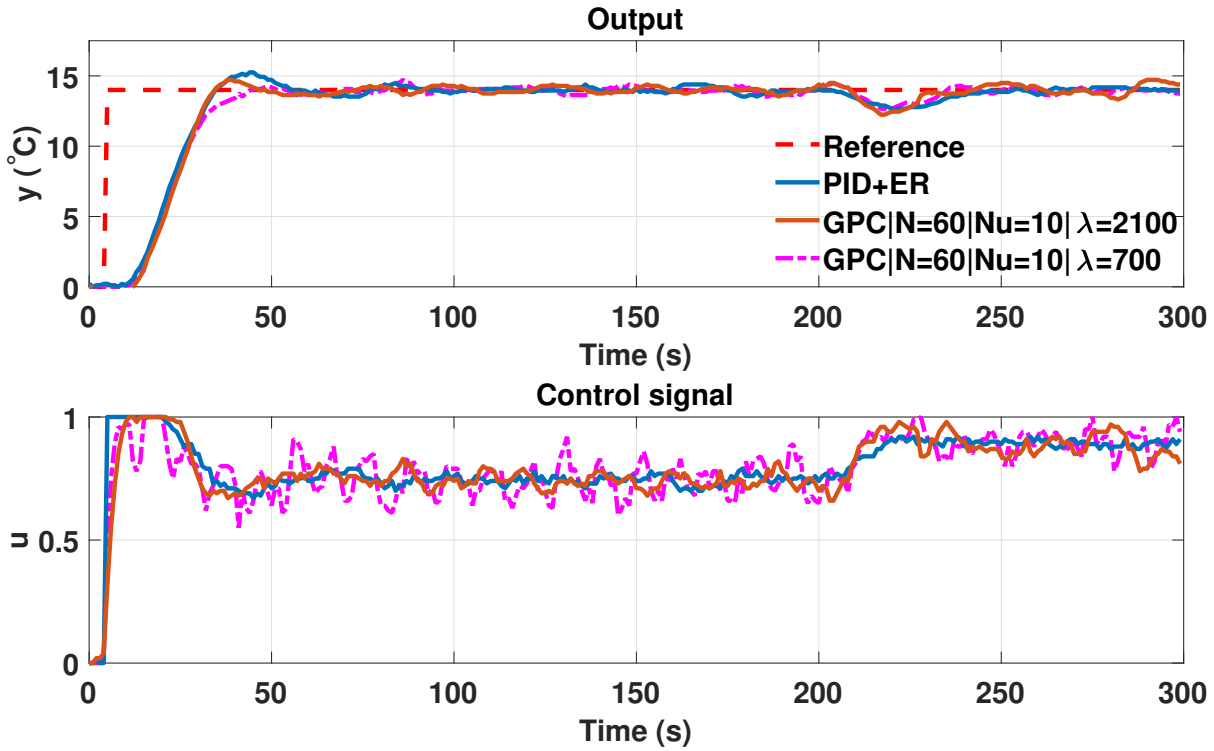


Figure 24 – Comparative analysis between PID and GPC performance for the experimental case

the improvement in GPC performance considering the second tuning is small compared to the other two approaches. Furthermore, the variability in the control action is even greater in the second tuning of the GPC due to the influence of the measurement noise on the control signal. The performance index obtained for the second GPC tuning was $J_{\text{GPC}} = 107.3$.

The results presented in this section show again that the choice between a more complex control strategy, such as MPC, instead of a PID, is directly related to the level of robustness required by the system. As shown, even for constrained processes, when a robust system is needed, the advantages of using an MPC, instead of a properly tuned PID with AW, are negligible or non-existent.

3.3 FINAL CONSIDERATIONS

This chapter presented a comparative study on PID, DTC and MPC methods, in terms of performance and robustness, when applied to dead-time processes. The objective of the study was to present information in order to facilitate the choice of the best strategy to be used, based on the characteristics of the process. First, it was shown that when the perfect model of the process is available, the FSP is able to present an ideal response for processes with dead time. Afterwards, an analysis is presented between a PID and an FSP for processes with dead time, and between a PID and a GPC for processes with dead time and constraints. In addition, an algorithm is proposed which allows the PID to handle processes subjected to different types of constraints.

The performance and robustness analysis between the PID and the FSP, for the unconstrained case, showed that, in most cases, both controllers present the same performance when robust tuning is considered. On the other hand, in cases where the available model effectively captures the process dynamics and tuning with a focus on fast performance is allowed, the FSP is capable of providing significantly better results than the PID.

For the analysis between the PID and the GPC, cases with characteristics commonly found in the industry were considered, such as modeling errors, dead time, constraints, and measurement noise. It has been shown that for processes with these characteristics, there are practically no advantages in using complex control strategies, such as MPC or DTC, instead of the well-known PID with AW, which is simple, effective, and presents a good trade-off between performance and robustness. The performance comparison for the case study, which considered measurement noise, showed that, for a given value of M_s, the performance of a PID with AW is significantly better than that provided by GPC and very similar to the one provided by DTC-GPC The analyzed experimental case, presented in Section 3.2.3, considered a process approximated by a FOPDT model subjected to constraints on the amplitude of the control signal. The results presented showed that, considering a robust tuning, the PID with AW presented a very similar performance in relation to GPC. However, considering a new tuning for the GPC, with a focus on performance, the improvement over the PID performance is practically negligible. This is caused by the limitation of GPC performance imposed by the available process model. In addition, it was observed that the control signal was significantly influenced by the measurement noise in the case considering a fast tuning for the GPC.

The main conclusion that can be drawn from this chapter is that the choice between a more complex control strategy or the classic PID is more dependent on the accuracy of the process model in effectively capturing the dynamics of the plant than on the absolute value of the dead time. For an industrial environment, where robust solutions are typically required, a PID with AW is capable of providing good or even better results than more complex strategies such as DTC or MPC. On the other hand, when robustness is not required and fast tuning is allowed, it is possible to obtain a significant performance improvement when using DTC or MPC strategies instead of PID, even for cases where the dead time is small.

4 PAPER 2 - CSPS: AN INTERACTIVE TOOL FOR CONTROL DESIGN AND ANALYSIS OF PROCESSES WITH INDUSTRIAL CHARACTERISTICS

This chapter covers the development and results of the paper "CSPS: an interactive tool for control design and analysis of processes with industrial characteristics" which was accepted and presented at the IFAC World Congress 2020. The study presents an user friendly interactive tool for control design, simulation and analysis of systems with characteristics commonly found in industry, such as dead time, constraints and measurement noise is presented. The tool is able to validate and compare, in a simple and intuitive way, the performance and robustness of the three control structures most widely used in industrial applications: PID, DTC, and MPC. Furthermore, the tool provides several options of techniques for handling input and output process constraints. A case study is used to illustrate some of the features of the tool.

4.1 TOOL DESCRIPTION

The constrained SISO-process simulator (CSPS) tool, shown in Figure 25, was developed using MATLAB®Graphic User Interface Design (GUIDE). CSPS tool allows the user to simulate SISO dead-time processes using PID, DTC and MPC controllers. Also, the tool considers three types of process constraints: saturation in magnitude and rate of change of control signal and in the output of the process. For dealing with process constraints the tool provides two options. The first one is to use an MPC strategy which handles process constraints by using an optimization procedure to find an optimal control action. The other option is to use anti-windup techniques, which are able to reduce the degradation of the closed-loop performance caused by the windup phenomenon (HIPPE, 2006).

The main properties of the CSPS tool are:

- user friendly graphical user interface for simulation and analysis of SISO processes with characteristics commonly found in industrial applications;
- performance analysis considering IAE performance index for setpoint tracking and load disturbance rejection;
- easy comparative analysis between different control strategies widely used in practice, such as PID, DTC, and MPC;
- performance analysis of different techniques used to handle process constraints;
- robustness analysis including important robustness measures.

The graphical interface is subdivided in five panels: *Process*, *Simulation*, *Constraints*, *Controller* and *Performance and robustness*. The functionalities of each panel are described in the next sections.

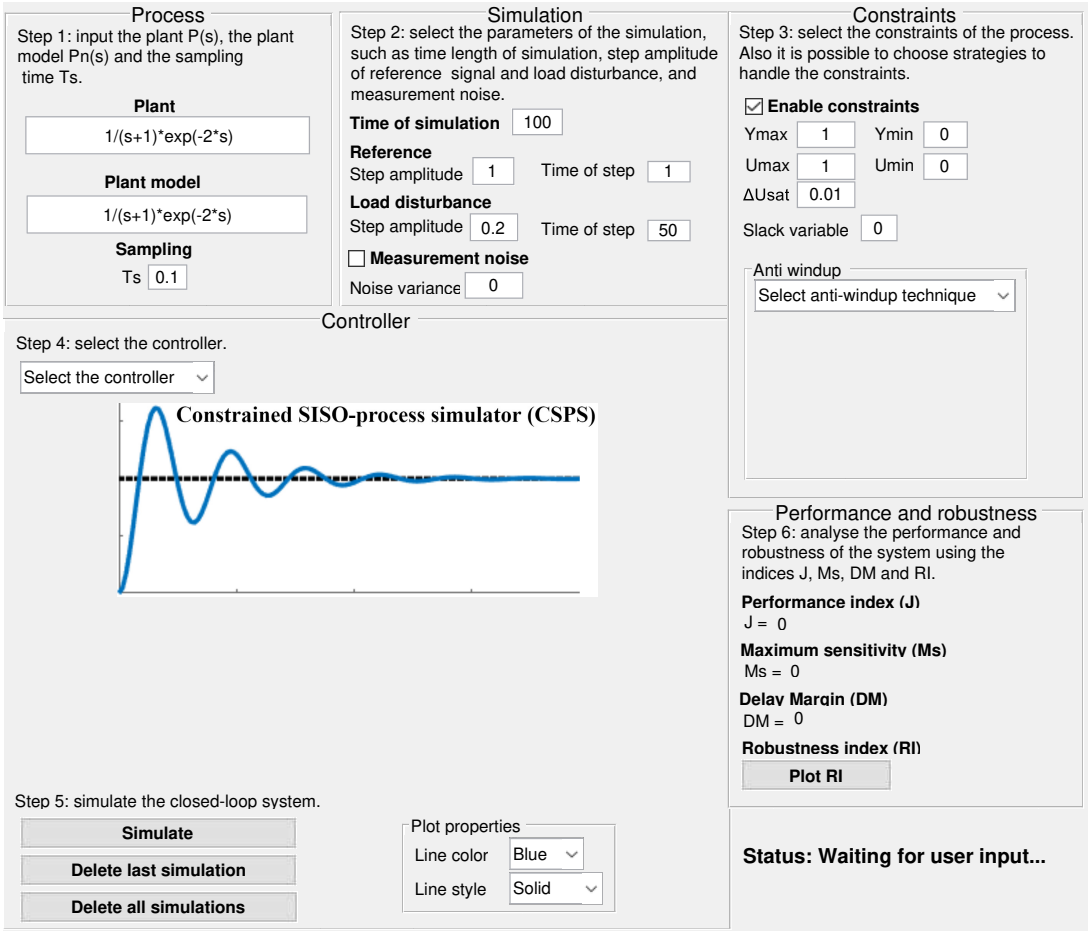


Figure 25 – Constrained SISO-process simulator (CSPS)

4.1.1 Process description

Firstly, in *Process* panel, the user must input a model which is the representation of the plant to be controlled, $P(s)$, and the model of the process, $P_n(s)$. The former is used as plant in all simulations, while the latter is used as plant model in the model-based approaches. Both models are represented as continuous-time transfer functions using variable s. For example, the plant model

$$P(s) = \frac{1}{s+1}e^{-2s}, \tag{4.1}$$

is input as `1/(s+1)*exp(-2*s)`.

The user must also input the sampling time, T_s, used for discretization of the models and controllers. All the process models are transformed into their discrete-time equivalents using the zero-order hold method, which assumes that the control signals are kept constant between two sampling instants. Despite some of the tuning rules available in the tool are defined in the continuous-time domain, all the controllers are discretized using the Tustin approximation technique and implemented in the discrete-time domain.

In *Simulation* panel, the user must input the duration of the simulation and the amplitude of the step used as reference signal. The tool also provides the option for considering a

step load disturbance and measurement noise with normal distribution and variance specified by the user.

4.1.2 Handling of constraints

CSPS tool provides the option to consider process constraints. For this purpose, it is necessary to mark the check box *enable constraints*, in *Constraints* panel, which then allows the user to define limits for the magnitude of control action (being $u_{\min}$ and $u_{\max}$ the minimum and maximum values, respectively), rate of change of control action (being Δu_{sat} the saturation limit) and magnitude of the output of the process (being $y_{\min}$ and $y_{\max}$ the minimum and maximum values, respectively).

For dealing with input constraints the tool provides three AW techniques widely used in practice for controllers which do not consider the constraints a priori (PID and DTC). The first one is the incremental algorithm (or velocity algorithm), presented in Section 2.3.1, which consists of calculating a control increment at each sampling period and adding to the previous control signal only the amount that does not saturate the actuator (ÅSTRÖM; WITTENMARK, 2013). This technique is widely used in industry for its simplicity of implementation in digital controllers. The second one is the back-calculation (BC) technique, presented in Section 2.3.2, which consists in adding an extra feedback signal to the input of the integrator, which is composed of the error between the output signal of the controller and the signal that is applied to the plant multiplied by a constant gain, T_t, known as tracking time parameter. The last one is the error recalculation (ER) technique, presented in Section 2.3.3 proposed in Flesch, Normey-Rico and Flesch (2017), which consists in modifying the current control signal and the current error signal to maintain the consistency between the control signal calculated by the controller and the input signal that is effectively applied to the plant.

For dealing with output constraints, the tool provides the option *constraints mapping*, which uses an approach based on the *clipping* technique, used in MPC strategies. The main idea of this technique is to calculate future output predictions using the process model, considering a prediction horizon, N, and compute a control action which guarantees that all output predictions are inside the region delimited by the constraints (see Silva, Flesch and Normey-Rico (2020) for details).

4.1.3 Available controllers and performance and robustness evaluation

CSPS tool provides three different control strategies for simulation: PID, DTC and MPC. These controllers can be select in *Controllers* panel. For the PID strategy, it is possible to use two different tunings, being the first one the approach proposed in (NORMEY-RICO; GUZMÁN, 2013), which uses only one tuning parameter, T_0. The second one is a manual tuning in which the user can freely choose the parameters k_c, T_i, T_d and α of a series PID with the structure defined in (2.17).

For the DTC option, which is implemented as an FSP, there are three tuning options. The first one is also based on the approach presented in Normey-Rico and Guzmán (2013) and uses T_0 as tuning parameter. The second tuning is an FSP in the discrete-time domain based on the GPC strategy, which can provide the same performance as the GPC for the unconstrained case (see Normey-Rico and Camacho (2007) for details). For this tuning, the user must specify three tuning parameters: prediction horizon, N, control horizon, N_u, and control increment weight, λ (tracking error weighting factor is assumed as $\delta = 1$). The last one is a manual tuning in which the user can freely set the parameters of a discrete FSP with reference filter, $F(z)$, primary controller, $C_{sp}(z)$, and robustness filter, $F_r(z)$.

For the MPC options, the tool provides two strategies: GPC and DTC-GPC. For the two strategies, the user must specify the tuning parameters N, N_u and λ. For the DTC-GPC option, the user must also specify the robustness filter, $F_r(z)$, which is used in the predictor structure. Both strategies use a quadratic programming solver provided by MATLAB, to find the optimal control action which satisfies all the constraints. If the optimization procedure results in an unfeasible solution, the tool will show a message indicating that a new tuning of the controller is necessary.

To quantify the performance of the closed-loop system CSPS tool uses a cost function, J, from Equation (2.75) which considers the integral of absolute error index (IAE) for setpoint tracking and load disturbance rejection. Just periods of time where the control signal can affect the process output due to the delay are considered in the cost function.

In terms of robustness analysis, CSPS tool provide three important indices: robustness index (R_I), delay margin (D_M) and maximum sensitivity (M_s), which were presented in equations (2.76), (2.77), (2.78), respectively.

4.2 CASE STUDY

In this section, an analysis of performance and robustness for a case study is presented to better illustrate the use of CSPS tool. The case study aims to explore the features of the proposed tool, not focusing on the performance or robustness of the controllers.

The plant considered in this section is a boiler, presented in Normey-Rico and Camacho (2007). The process is described by the linear model

$$P(s) = \frac{2e^{-5s}}{s(s+1)(0.5s+1)(0.1s+1)}. \tag{4.2}$$

For tuning the controllers, the dynamics of the process were approximated by an IPDT model given by

$$P_n(s) = \frac{2}{s}e^{-6.5s}, \tag{4.3}$$

with time given in minutes. The sampling time used for simulating the process and discretizing the controllers is $T_s = 0.5\,\text{min}$. The first simulation does not take into account measurement noise and considers constraints in magnitude and rate of change of the control signal, $u_{\min} =$

-0.05, $u_{\max} = 0.05$, $\Delta u_{\text{sat}} = 0.01$, and in the output of the process, $y_{\min} = 0$ and $y_{\max} = 1.1$. In this case a PID controller is used and it is tuned for fast performance with $M_s = 4.6$ using the rule presented in Section 2.1.3, with a closed loop time constant of $T_0 = 4\,\text{min}$, resulting in a controller $C_{\text{PID}}(s)$ and a reference filter $F(s)$ given by

$$C_{\text{PID}}(s) = \frac{0.45(s+0.30)(s+0.06)}{s(s+1.71)}, \tag{4.4}$$

$$F(s) = \frac{0.27(s+0.25)}{s+0.06}. \tag{4.5}$$

Figure 26 shows the simulation of the closed-loop system for a unit step reference at $t = 1\,\text{min}$ and a load disturbance of amplitude -0.04 at $t = 100\,\text{min}$, considering three cases: without AW; with ER AW; and with ER AW and constraints mapping (CM), which is used to handle output constraints, with a prediction horizon of $N = 6$.

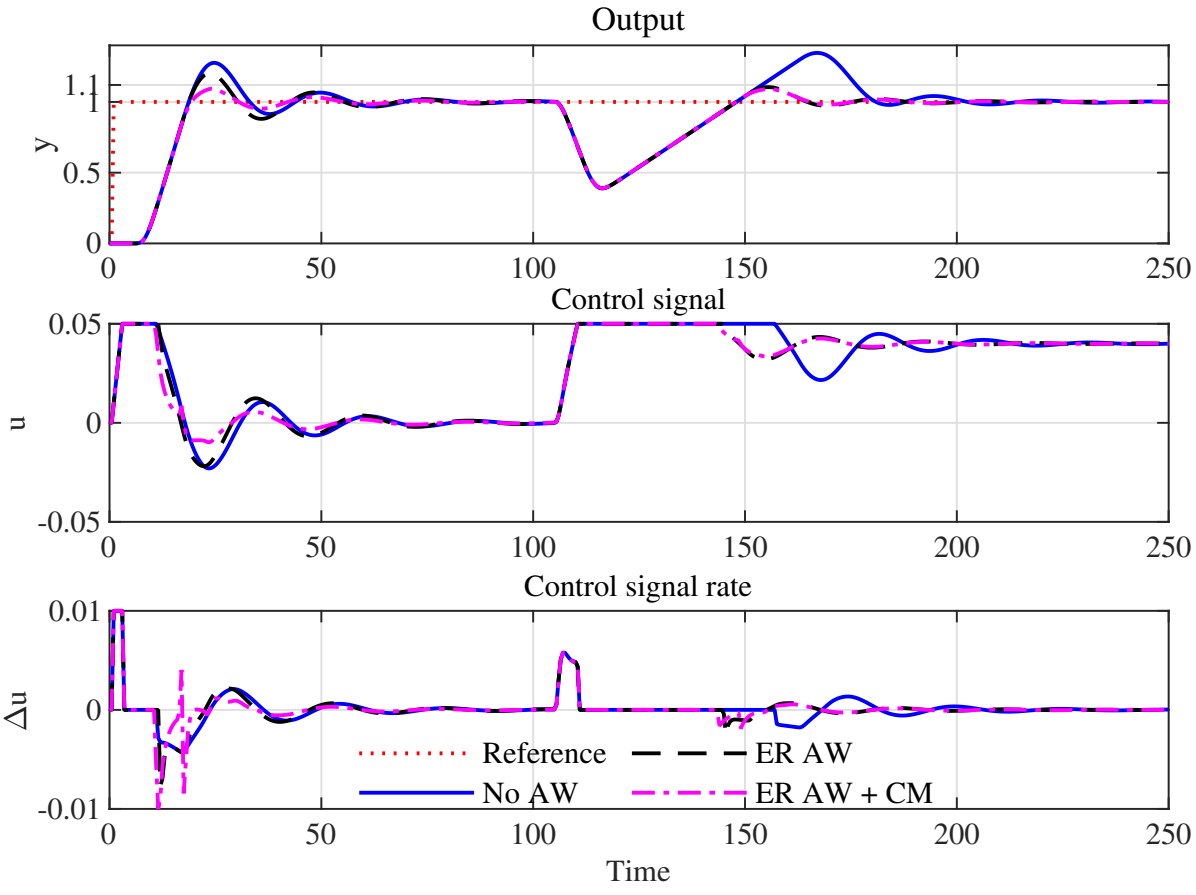


Figure 26 – Closed-loop performance of the integrating case without measurement noise

As can be seen in Figure 26, the case without AW presented high overshoot and oscillations, caused by saturation constraints, resulting in a performance index of $J_{\text{NOAW}} = 32.65$. The PID controller with ER AW was able to reduce overshoot and oscillations, providing a better performance when compared to the case without AW and resulting in $J_{\text{ER}} = 26.65$. The last case presented the best performance, being able to reduce the overshoot and also able to deal

with all the constraints considered, resulting in a performance index of $J_{\text{ER+CM}} = 24.42$. The obtained value for D_M in this case is $D_{M_{\text{PID}}} = 1.87\,\text{min}$.

The second simulation of the system considers measurement noise with normal distribution and variance of 0.03. In this case, the performance of FSP without AW, FSP with incremental algorithm (IA) AW and DTC-GPC are compared. Both controllers are tuned for a robust solution with $M_s = 2.0$. The FSP was tuned considering $T_0 = 8\,\text{s}$, resulting in a primary controller, $C_{sp}(s) = 0.06$, and robustness filter

$$F_r(s) = \frac{2.81(s+0.04)}{s+0.12}. \tag{4.6}$$

The DTC-GPC was tuned with $N = 40$, $N_u = 12$, $\lambda = 15 \times 10^3$ and a discrete-time robustness filter

$$F_r(z) = \frac{1.13z - 1.12}{z - 0.99}. \tag{4.7}$$

Furthermore, in this case, the output constraints were relaxed in DTC-GPC to avoid infeasibility of the optimization procedure, due to noisy measurements.

Figure 27 shows the simulation of the system considering measurement noise. As can be seen, the FSP with IA presented the best performance when compared to the other two cases, with no overshoot and fast disturbance rejection response. The performance indices obtained for this case for FSP without AW, FSP with AW and DTC-GPC are $J_{\text{FSP-NOAW}} = 35.1$, $J_{\text{FSP-IA}} = 32.1$ and $J_{\text{DTC-GPC}} = 52.7$, respectively. Both controllers presented similar robustness properties in terms of dead time uncertainties, presenting $D_{M_{\text{FSP}}} = 5.24\,\text{min}$ and $D_{M_{\text{DTC-GPC}}} = 5.71\,\text{min}$.

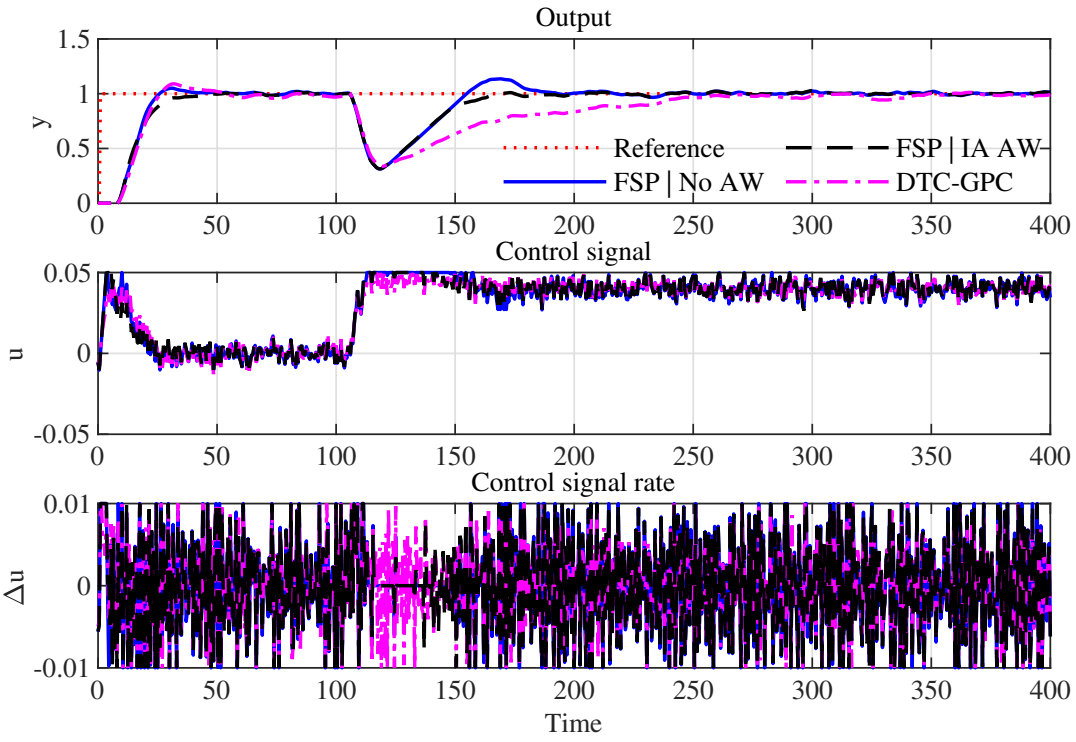


Figure 27 – Closed-loop performance of the integrating case with measurement noise

In Figure 28 the robustness index, $R_I(\omega)$, of the three controllers and the modeling error, $\delta P(\omega)$, of the process are shown. As can be seen, the robustness properties of FSP and DTC-GPC were very similar. On the other hand, as expected, the PID tuned for fast performance was considerably less robust when compared to the other two strategies.

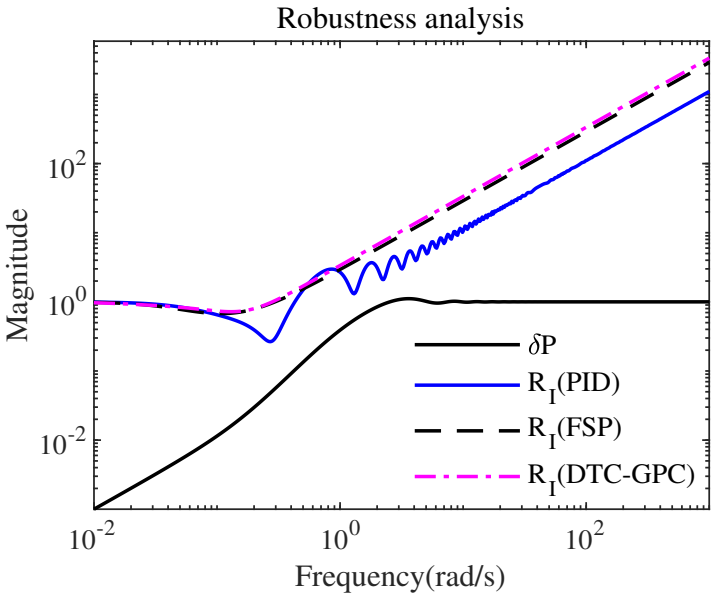


Figure 28 – Robustness analysis

All the results of the performance and robustness analysis presented in this case study were easily obtained by using the proposed tool. In addition, the tuning procedure of the controllers is easy and intuitive, and the performance and robustness comparison between different control structures can be done in a simple way.

4.3 FINAL CONSIDERATIONS

In this chapter, the development of an user friendly interactive tool with graphical interface for simulation and analysis of SISO processes with dead time and constraints is described. The main features of the tool include: easy tuning of controllers widely used in industry; performance analysis of the closed-loop response of processes including characteristics commonly found in practical applications; possibility to include anti-windup action in the structure of the controllers; robustness analysis including important robustness indices. These characteristics make the proposed tool a good option for teaching important concepts of control engineering. Furthermore, the tool can be used to decide the best control strategy to be used based on the characteristics of the process. A case study considering an integrating process was presented for a better illustration of the features of the tool. The CSPS tool is available for download at http://rodolfoflesch.prof.ufsc.br/cspstool.

5 PAPER 3 - PID DESIGN METHOD BASED ON GPC WITH INPUT CONSTRAINTS HANDLING

The current chapter aims to outline the development and results of the research paper entitled "PID Design Method Based on GPC with Input Constraints Handling," which was submitted to journal of ISA Transactions. The study presents a PID tuning approach that incorporates an additional scheme referred to as the Control Signal Governor (CSG) method, designed to control first or second order, stable, integrative, or unstable processes subjected to input constraints. The methodology demonstrates equivalent performance when compared to the traditional GPC approach. The CSG scheme, responsible for handling input constraints, calculates the same optimal solution as the constrained GPC approach, but without the need for an online optimizer, turning the method suitable for processes with fast dynamics or for cases where only a low-cost microcontroller is available. The proposed method was evaluated through a simulation of a second-order underdamped process subjected to input constraints. The performance and computational execution time of the control action of the proposed approach and the traditional GPC were compared using the MATLAB® software. The main contributions of this paper are:

- the introduction of a PID tuning approach incorporating the CSG structure for input constraints handling;
- a PID controller with the same optimal performance as presented by the traditional GPC even in constrained cases;
- the elimination of the necessity of an online optimizer;
- improved computational efficiency in terms of execution time to compute the control action when compared to the GPC.

5.1 GEOMETRIC REPRESENTATION OF THE FEASIBLE REGION

This section aims to present the geometric representation of the input constraints (magnitude and increment of control action) and also of the cost function of the QP problem associated with the GPC in order to give some insights about the proposed approach. In all cases, the geometric representation considers the hyperspace of the control increments as axes, so a control horizon of $N_u = 2$ is considered in this section in order to facilitate the visualization of the geometric interpretation of the constraints.

5.1.1 Geometric form of input constraints

Based on the inequalities presented in Equation (2.41), an arbitrary example of the geometric representation of the constraints in the $\Delta u(k) \times \Delta u(k+1)$ plane, for a control horizon

$N_u = 2$, is presented in Figure 29. Polytope A is defined by the increment constraints, polytope B is defined by the magnitude constraints, and polytope C is obtained by the intersection of the areas defined the constraints individually, which also represents the feasible region of the problem.

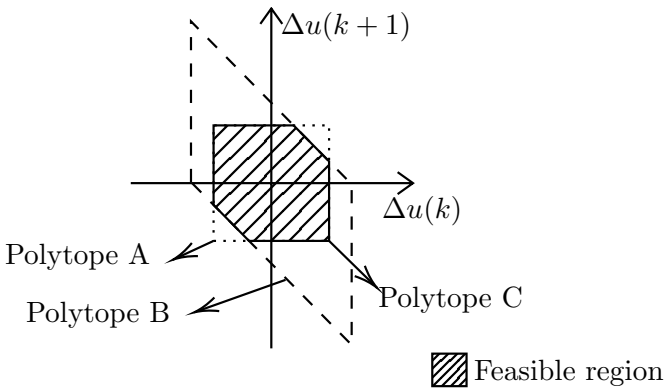


Figure 29 – Convex polytope for constraints in magnitude and increment of control action

Because the second inequality in Equation (2.41) has the term $u(k-1)$, the position of the polytopes B and C in the plane may change at each sampling instant k, even if the absolute limits of the control action remain unchanged.

5.1.2 Geometric interpretation of the QP problem

Consider Equation (2.36) with $J_{\text{MPC}} = c$, where $c \in \mathbb{R}$. For different values of c, Equation (2.36) defines level curves which can be geometrically interpreted as ellipsoids of particular sizes in $\mathbb{R}^{N_u}$ with center at $\Delta\mathbf{u} = -\mathbf{H}^{-1}\mathbf{b}$ (which represents the optimal solution for the unconstrained case).

For the constrained case, the minimization of (2.36) may be interpreted as finding the point at which the smallest ellipsoid touches the polytope of the feasible region (SERON; DONA; GOODWIN, 2000). For a better illustration of the previous affirmation, Figure 30 shows the geometric representation of the optimal solution of the QP problem for a control horizon of $N_u = 2$ considering only constraints in the increment of control action, where $\Delta\mathbf{u}^{\star}_{\text{UC}}$ is the optimal solution for the unconstrained case and $\Delta\mathbf{u}^{\star}$ is the optimal solution for the constrained case.

From Figure 30 it is also possible to observe that, for the particular case with $N_u = 1$, the optimal solution for the constrained case, $\Delta u(k)^{\star}$, will be in the x axis of the plane and can be obtained from the saturation of the unconstrained optimal solution $\Delta u(k)^{\star} = \text{sat}(\Delta u^{\star}_{\text{UC}}(k), \Delta u_{\min}, \Delta u_{\max})$, where the generalized saturation function, $\text{sat}(m, n_{\min}, n_{\max})$ is defined as

$$\text{sat}(m, n_{\min}, n_{\max}) = \begin{cases} n_{\min} & \text{if} \quad m \leq n_{\min} \\ m & \text{if} \quad n_{\min} < m < n_{\max} \\ n_{\max} & \text{if} \quad m \geq n_{\max}, \end{cases} \tag{5.1}$$

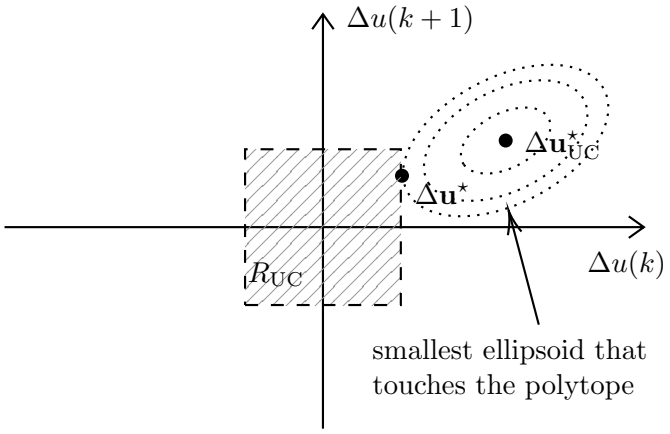


Figure 30 – Geometric interpretation of the optimization problem for the constrained case considering $N_u = 2$

where $n_{\min}$ and $n_{\max}$ define the lower and upper saturation limits, respectively.

As discussed in (SERON; DONA; GOODWIN, 2000), it is possible to apply a linear transformation

$$\begin{aligned} &T : \mathbb{R}^{N_u} \rightarrow \mathbb{R}^{N_u} \\ &T(\Delta \mathbf{u}) = \Delta \tilde{\mathbf{u}} = \mathbf{H}^{1/2} \Delta \mathbf{u}, \end{aligned} \tag{5.2}$$

in order to map the coordinate system from $\Delta \mathbf{u}$ to $\Delta \tilde{\mathbf{u}}$. In this way, for the particular case of $N_u = 2$, the ellipsoids in the $\Delta u(k) \times \Delta u(k+1)$ plane can be represented in the $\Delta \tilde{u}(k) \times \Delta \tilde{u}(k+1)$ plane as circumferences with center at $\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}} = \mathbf{H}^{1/2} \Delta \mathbf{u}^{\star}_{\mathrm{UC}} = -\mathbf{H}^{-1/2} \mathbf{b}$ (which represents the optimal solution for the unconstrained case in the transformed coordinates). The new feasible region, $\tilde{R}_{\mathrm{UC}}$, can be obtained from the following inequalities

$$\tilde{\mathbf{A}}_{\mathbf{c}} \Delta \tilde{\mathbf{u}} \leq \mathbf{b}_{\mathbf{c}}, \tag{5.3}$$

where $\tilde{\mathbf{A}}_{\mathbf{c}} = \mathbf{A}_{\mathbf{c}} \mathbf{H}^{-1/2}$.

In the transformed coordinates, the minimization problem with constraints comes down to finding the smallest circumference that touches the polytope of region $\tilde{R}_{\mathrm{UC}}$, which is equivalent to finding the point in the polytope which presents the least Euclidean distance to the point $\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}$. Figure 31 presents the geometric interpretation for the minimization problem in the transformed coordinates.

Notice that for the particular case of $N_u = 2$, in case the line which contains the closest point to $\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}$ is known a priori, the point in the polytope which presents the least Euclidean distance can be obtained from an orthogonal projection of the point $\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}$ to that line.

Consider that the subspace S_i which contains the closest point to $\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}$, in the modified coordinates, is given by $S_i = \{\Delta \tilde{\mathbf{u}} \,|\, \tilde{\mathbf{a}}_i \Delta \tilde{\mathbf{u}} = b_i\} \in \mathbb{R}^{N_u - 1}$, where $\tilde{\mathbf{a}}_i$ is the i^{th} row of matrix $\tilde{\mathbf{A}}_{\mathbf{c}}$ and b_i is the i^{th} element of vector $\mathbf{b}_{\mathbf{c}}$. In this case, the orthogonal projection $\mathrm{proj}_{S_i}(\tilde{\mathbf{u}}^{\star}_{\mathrm{UC}})$ of the point $\tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}$ to the subspace S_i can be obtained as (MEYER, 2000)

$$\mathrm{proj}_{S_i}(\Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}}) = \Delta \tilde{\mathbf{u}}^{\star} = \left[\mathbf{I} - \tilde{\mathbf{a}}_i^T (\tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T)^{-1} \tilde{\mathbf{a}}_i\right] \Delta \tilde{\mathbf{u}}^{\star}_{\mathrm{UC}} + \tilde{\mathbf{a}}_i^T (\tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T)^{-1} b_i. \tag{5.4}$$

In order to find the optimal point in the original coordinates, i.e. $\Delta \mathbf{u}^{\star}$, firstly, Equation (5.4) must be substituted in Equation (5.2), considering that $\tilde{\mathbf{a}}_i = \mathbf{a}_i \mathbf{H}^{-1/2}$, which results in

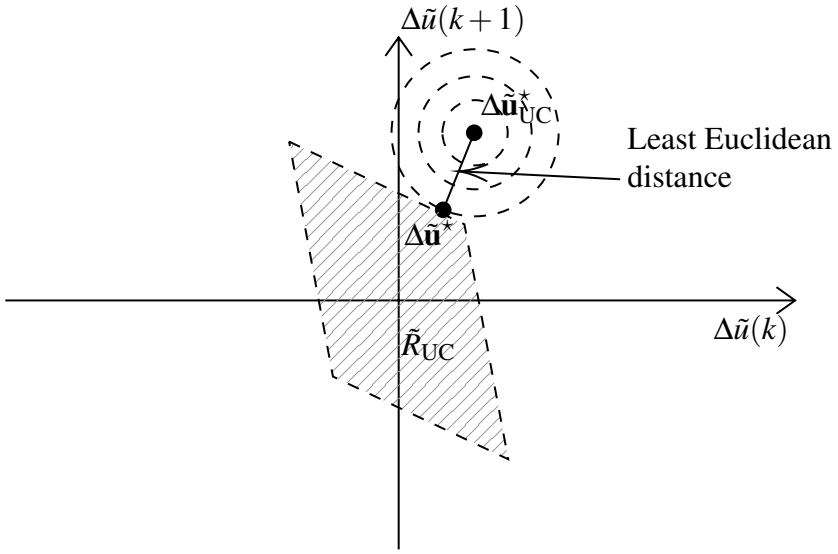


Figure 31 – Geometric interpretation of the minimization problem with constraints for $N_u = 2$ in the transformed coordinates

$$\mathbf{H}^{1/2}\Delta\mathbf{u}^{\star} = \left[\mathbf{I} - \mathbf{D}\mathbf{a}_i\mathbf{H}^{-1/2}\right]\mathbf{H}^{1/2}\Delta\mathbf{u}^{\star}_{\mathrm{UC}} + \mathbf{D}b_i, \tag{5.5}$$

where $\mathbf{D}$ is defined as

$$\mathbf{D} = (\mathbf{a}_i\mathbf{H}^{-1/2})^T(\mathbf{a}_i\mathbf{H}^{-1/2}(\mathbf{a}_i\mathbf{H}^{-1/2})^T)^{-1}. \tag{5.6}$$

After isolating the term $\Delta\mathbf{u}^{\star}$, Equation (5.5) can be rewritten as

$$\Delta\mathbf{u}^{\star} = \left[\mathbf{I} - \frac{\mathbf{H^{-1}}\mathbf{a}_i^T\mathbf{a}_i}{\mathbf{a}_i\mathbf{H^{-1}}\mathbf{a}_i^T}\right]\Delta\mathbf{u}^{\star}_{\mathrm{UC}} + \frac{\mathbf{H^{-1}}\mathbf{a}_i^T b_i}{\mathbf{a}_i\mathbf{H^{-1}}\mathbf{a}_i^T}, \tag{5.7}$$

which divides the constrained optimal solution in a term which is function of the unconstrained optimal solution, $\Delta\mathbf{u}^{\star}_{\mathrm{UC}} = -\mathbf{H}^{-1}\mathbf{b}$, and another independent term, which is function of the element b_i of vector $\mathbf{b_c}$. Only the control increment associated with the current instant, $\Delta u(k)$, is applied to the plant, therefore only the first row of each term of Equation (5.7) needs to be computed. The presented solution is valid for any kind or combination of input constraints and size of control horizon. However, for large values of control horizon and considering many constraints, the identification of the subspace in which the unconstrained solution must be projected may become very complex.

Based on the interpretation discussed in this section, an analytical solution for the constrained optimization problem, considering a short control horizon of two steps, is presented in Section 5.2.

5.2 ANALYTICAL SOLUTION FOR A CONTROL HORIZON $N_u = 2$

For the particular case of a control horizon $N_u = 2$, the constrained optimal solution presented in Equation (5.7) can be interpreted as the point at which the projection of the unconstrained optimal solution, $\Delta\mathbf{u}^{\star}_{\mathrm{UC}}$, performed in a direction defined by the vector formed by the

first row of matrix $\mathbf{V}$, where

$$\mathbf{V} = \left[\mathbf{I} - \frac{\mathbf{H^{-1}} \mathbf{a}_i^T \mathbf{a}_i}{\mathbf{a}_i \mathbf{H^{-1}} \mathbf{a}_i^T} \right], \tag{5.8}$$

touches the edge of the polytope R_{UC}.

Based on this interpretation, this sections aims to present an analytical solution for the optimization problem considering constraints in magnitude and increment of control action for the particular case of a control horizon $N_u = 2$.

5.2.1 Analytical solution considering increment constraints

In cases in which only the constraint associated with the first increment of the control signal, $\Delta u(k)$, is violated, that is $\Delta u^{\star}_{\mathrm{UC}}(k) > \Delta u_{\max}$ or $\Delta u^{\star}_{\mathrm{UC}}(k) < \Delta u_{\min}$, with $\Delta u^{\star}_{\mathrm{UC}}(k+1)$ in the feasible region, the constrained optimal solution is equal to the unconstrained optimal solution saturated, i.e $\Delta u^{\star}(k) = \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k), \Delta u_{\min}, \Delta u_{\max})$.

For the cases in which the second increment of control action, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, violates any of the constraints, the constrained optimal solution can be obtained based on the projection of the unconstrained solution $\Delta u^{\star}_{\mathrm{UC}}(k+1)$. Firstly, it is necessary to obtain the direction in which $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$ must be projected. This direction is defined by the vector $\mathbf{v} = [v_a, v_b]$, which can be obtained from the first row of matrix $\mathbf{V}$ for $i = 2$ or $i = 4$. As the lines defined by the constraints for $i = 2$ and $i = 4$ are parallel to each other, they have the same slope. Summarizing, it is possible to obtain the optimal solution for the constrained problem, $\Delta \mathbf{u}^{\star}$, by calculating the point where the line which is normal to the vector $\mathbf{v}$, and passes through the point $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$, touches the face of the polytope related to the active constraint.

In order to facilitate the development of the analytical solution for the optimization problem considering a control horizon $N_u = 2$, a generalized case with constraints only in the increment of control signal is considered. The generalized case is represented in Figure 32, in which, for an instant k, the unconstrained optimal solution, $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$, is found at a specific region of the $\Delta u(k) \times \Delta u(k+1)$ plane.

In Figure 32 it can be seen that only the second control increment of vector $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, is outside the limit defined by the upper constraint, $\Delta u_{\max}$. Based on this fact, it is possible to obtain the line in which $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$ should be projected. This line, represented in the Figure 32 as l_1, has its slope defined by the vector $\mathbf{v}$. Thus, it is possible to obtain the optimal solution of the constrained problem, $\Delta \mathbf{u}^{\star}$, by calculating the point where the line normal to l_1 passes through the point $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$, represented in Figure 32 by l_2, touches the face of the polytope related to the active constraint, defined by the line $\Delta u(k+1) = \Delta u_{\max}$ in this particular case. In practical terms, as only the first increment of the optimal control signal, $\Delta u^{\star}(k)$, is required for implementation, only the $\Delta u(k)$ coordinate of the point $\Delta \mathbf{u}^{\star}$ needs to be calculated.

The computation of $\Delta u^{\star}(k)$ can be performed analytically as follows. Line l_2 is given by

$$\Delta u(k+1) = \gamma \Delta u(k) + \psi, \tag{5.9}$$

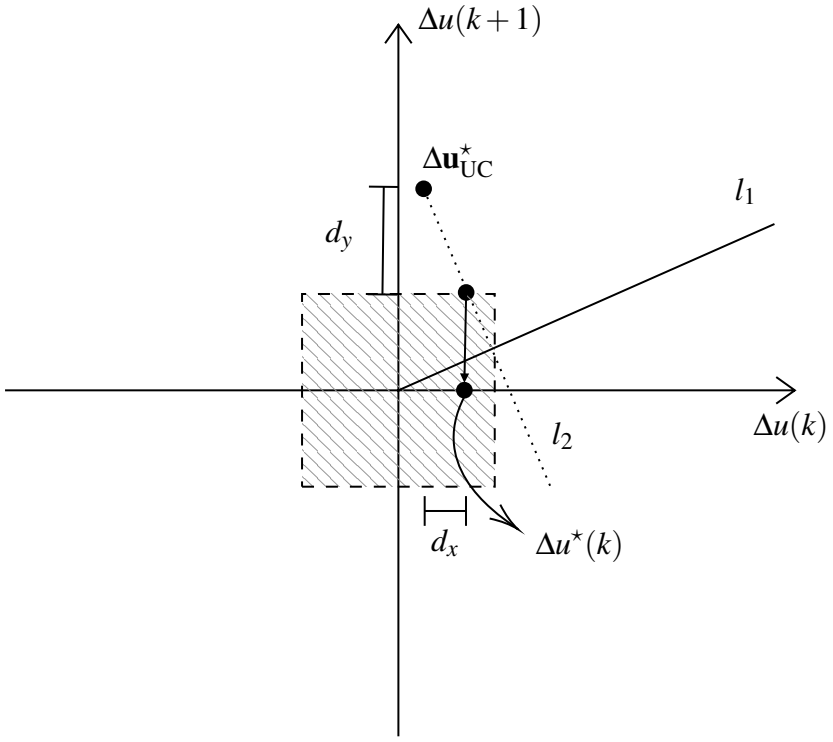


Figure 32 – Geometric interpretation for a generalized case with constraints in the increment of control action for $N_u = 2$

where γ is the slope and ψ is the intercept. As l_2 is normal to l_1, the slope of l_2, γ, is given by

$$\gamma = -v_a/v_b. \tag{5.10}$$

Furthermore, as l_2 passes through the point $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$, it is possible to compute ψ as

$$\psi = \Delta u^{\star}_{\mathrm{UC}}(k+1) - \gamma \Delta u^{\star}_{\mathrm{UC}}(k). \tag{5.11}$$

Finally, substituting (5.11) in Equation (5.9), the line l_2 can be described as

$$\Delta u(k+1) = \gamma \Delta u(k) + \Delta u^{\star}_{\mathrm{UC}}(k+1) - \gamma \Delta u^{\star}_{\mathrm{UC}}(k). \tag{5.12}$$

From Equation (5.12), it is possible to obtain $\Delta u^{\star}(k)$ by substituting $\Delta u(k+1)$ by the line which defines the active constraint, being it defined as $\Delta u(k+1) = \Delta u_{\max}$ for the particular case shown in Figure 32, resulting in

$$\Delta u^{\star}(k) = \mathrm{sat}\left(\Delta u^{\star}_{\mathrm{UC}}(k) + U_{\mathrm{aux}_1}, \Delta u_{\min}, \Delta u_{\max}\right), \tag{5.13}$$

where

$$U_{\mathrm{aux}_1} = \frac{-\Delta u^{\star}_{\mathrm{UC}}(k+1) + \Delta u_{\max}}{\gamma}. \tag{5.14}$$

The saturation function is used to guarantee that the optimal solution of the constrained problem is feasible for the cases in which the projection of the point $\Delta \mathbf{u}^{\star}_{\mathrm{UC}}$ to the line l_1 does not touch the polytope.

In order to avoid the need to identify which constraint is active at each sampling instant, it is possible to perform a procedure to generalize the method from the geometric interpretation of the problem, based on Figure 32. In Figure 32, d_y represents the value which must be subtracted from $\Delta u^{\star}_{\mathrm{UC}}(k+1)$ so that the resulting point lies on the line which defines the active constraint, and d_x defines the correction that must be applied to $\Delta u^{\star}_{\mathrm{UC}}(k)$ to obtain $\Delta u^{\star}(k)$. For the generalized case, d_y can be computed as

$$d_y = \Delta u^{\star}_{\mathrm{UC}}(k+1) - \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k+1), \Delta u_{\min}, \Delta u_{\max}). \tag{5.15}$$

Therefore, substituting

$$\Delta u(k+1) = \Delta u^{\star}_{\mathrm{UC}}(k+1) - d_y \tag{5.16}$$

in Equation (5.12), $\Delta u^{\star}(k)$ can be computed as

$$\Delta u^{\star}(k) = \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k) + d_x, \Delta u_{\min}, \Delta u_{\max}), \tag{5.17}$$

where

$$d_x = \frac{\Delta u^{\star}_{\mathrm{UC}}(k+1) - \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k+1), \Delta u_{\min}, \Delta u_{\max})}{-\gamma}. \tag{5.18}$$

The formulation above can be represented as a block diagram, as shown in Figure 33, where $\mathrm{sat}_1(\cdot)$ and $\mathrm{sat}_2(\cdot)$ are saturation blocks with lower and upper saturation limits of $\Delta u_{\min}$ and $\Delta u_{\max}$, and $K_f = -\gamma$. The sector of the structure which is highlighted in Figure 33 (dotted line) will be named control signal governor (CSG), since the proposed method presents some similarities when compared to error and reference governor methods. Even though there are also some similarities with anti-windup approaches, we avoid using this term because the proposed method can modify the control action even if the unconstrained solution obtained by the GPC, $\Delta u^{\star}_{\mathrm{UC}}(k)$, is within the bounds defined by the constraints. This may happen if only $\Delta u^{\star}_{\mathrm{UC}}(k+1)$ violates at least one of the constraints.

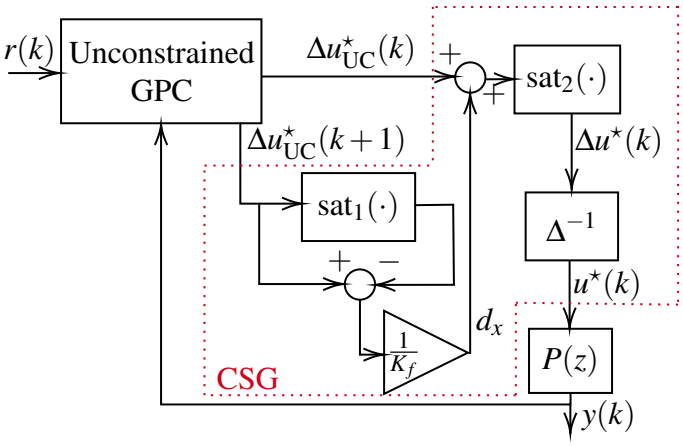


Figure 33 – Block diagram for the calculation of the optimal control action considering constraints in increment and $N_u = 2$

The methodology discussed above can be represented as a pseudo-code form, as shown in Algorithm 5. It is important to notice that all the steps can be solved analytically, since the

unconstrained version of GPC has analytical solution and the other steps can be represented as fundamental block diagram operations, as shown in Figure 33.

Algorithm 5: Fast computing of the optimal solution considering constraints in increment of control action

```
initialize variables;
compute vector v (first row of V for i = 2 as in (5.8));
compute γ as in (5.10);
repeat
    compute Δu*_UC as in (2.38);
    compute d_x as in (5.18);
    compute Δu*(k) as in (5.17);
    u*(k) ← u(k−1) + Δu*(k);
    apply u*(k) to the plant;
    update the variables;
    k ← k+1;
    wait T_s;
until controller is stopped;
```

5.2.2 Analytical solution considering magnitude constraints

In the case where only control magnitude constraints are considered, the approach used consists in verifying if the second control action computed, $u(k+1)$ (or represented using the increment of control action as $u(k-1)+\Delta u(k)+\Delta u(k+1)$), violates any of the constraints considered, and, in such case, find out the most violated constraint, which is referred in this paper as the main active constraint. As mentioned in Section 5.2, this verification can also be performed based on the unconstrained optimal solution, $\Delta\mathbf{u}^{\star}_{\mathrm{UC}}$, by verifying if

$$\Delta u^{\star}_{\mathrm{UC}}(k+1) > u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k), \tag{5.19}$$

or if

$$\Delta u^{\star}_{\mathrm{UC}}(k+1) < u_{\min} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k). \tag{5.20}$$

The following generalized case is presented in order to illustrate the method discussed previously and to present the analytical solution for the constrained optimization problem. Consider the case with lower and upper magnitude saturation limits where at a time instant k the unconstrained increment of the control signal, $\Delta\mathbf{u}^{\star}_{\mathrm{UC}}$, is found at a specific region of the $\Delta u(k)\times\Delta u(k+1)$ plane, as shown in Figure 34.

The same analysis based on Figure 32 can be extended for the case which considers constraints in magnitude of control action. However, in order to simplify the geometric interpretation and facilitate the development of the analytical solution, a new coordinate system, considering a $\tilde{x}\times\tilde{y}$ plane, is presented in Figure 34. In the transformed coordinate system, the line which defines the active constraint (which is $\Delta u(k+1)=u_{\max}-u(k-1)-\Delta u(k)$ in this particular case) is represented as $\tilde{y}=-\tilde{x}$, because in this system the line passes through the origin.

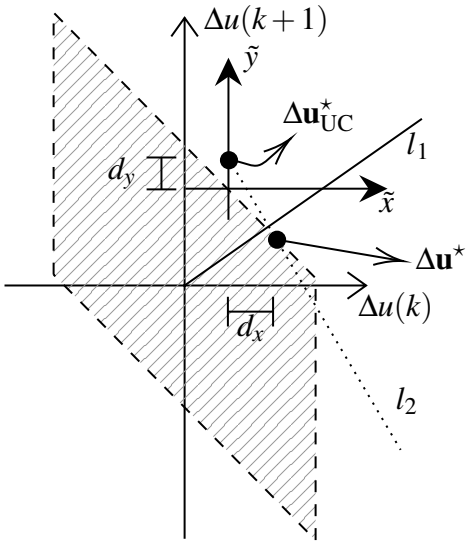


Figure 34 – Projection of the unconstrained optimal solution for the case considering magnitude constraints for $N_u = 2$

The line l_2 is represented as $\tilde{y} = \gamma\tilde{x} + d_y$, where $d_y = \Delta u^{\star}_{\mathrm{UC}}(k+1) - (u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k))$. The coordinate $\tilde{x}$ of the intersection point of the two lines can be computed as

$$\tilde{x} = d_x = \frac{d_y}{-\gamma - 1}. \tag{5.21}$$

To avoid the need for identifying which constraint is active at each sampling instant for the cases considering lower and upper magnitude saturation limits, d_y can be obtained as

$$d_y = \Delta u^{\star}_{\mathrm{UC}}(k+1) - \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k+1), U_{1_{\min}}, U_{1_{\max}}), \tag{5.22}$$

where

$$\begin{aligned} U_{1_{\min}} &= u_{\min} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k), \\ U_{1_{\max}} &= u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k). \end{aligned} \tag{5.23}$$

Finally, the constrained optimal solution, $\Delta u^{\star}(k)$, can be written as

$$\Delta u^{\star}(k) = \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k) + d_x, u_{\min} - u(k-1), u_{\max} - u(k-1)), \tag{5.24}$$

where d_x can be computed as

$$d_x = \frac{\Delta u^{\star}_{\mathrm{UC}}(k+1) - \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k+1), U_{1_{\min}}, U_{1_{\max}})}{-\gamma - 1}. \tag{5.25}$$

The method discussed previously can be represented as the same block diagram shown in Figure 33, therefore considering that $\mathrm{sat}_1(\cdot)$ has lower and upper saturation limits of $U_{1\min}$ and $U_{1\max}$, respectively, $\mathrm{sat}_2(\cdot)$ has lower and upper saturation limits of $u_{\min} - u(k-1)$ and $u_{\max} - u(k-1)$, respectively, and $K_f = -\gamma - 1$.

The above discussion is also presented in a pseudo-code form, as shown in Algorithm 6.

Algorithm 6: Fast computing of the optimal solution considering constraints in magnitude of control action

initialize variables;
compute vector $\mathbf{v}$ (first row of $\mathbf{V}$ for $i = 2$ as in (5.8));
compute γ as in (5.10);
repeat
  compute $\Delta\mathbf{u}^{\star}_{\mathrm{UC}}$ as in (2.38);
  compute $U_{1_{\min}}$ and $U_{1_{\max}}$ as in (5.23);
  compute d_x as in (5.25);
  compute $\Delta u^{\star}(k)$ as in (5.24);
  $u^{\star}(k) \leftarrow u(k-1) + \Delta u^{\star}(k)$;
  apply $u^{\star}(k)$ to the plant;
  update the variables;
  $k \leftarrow k+1$;
  wait T_s;
until *controller is stopped*;

5.2.3 Analytical solution considering both increment and magnitude constraints

In the case constraints both in increment and magnitude of the manipulated variable are considered, the polytope formed by such constraints is not constant, so the functions $\min(\cdot)$ and $\max(\cdot)$ are used to verify if the unconstrained optimal solution is outside the feasible region. For the case in which only the first increment of control action computed, $\Delta u^{\star}_{\mathrm{UC}}(k)$, violates any of the constraints, i.e., if

$$\Delta u^{\star}_{\mathrm{UC}}(k) > \min\{\Delta u_{\max}, \Delta u_{\max} - u(k-1)\}, \tag{5.26}$$

or if

$$\Delta u^{\star}_{\mathrm{UC}}(k) < \max\{\Delta u_{\min}, \Delta u_{\min} - u(k-1)\}, \tag{5.27}$$

with $\Delta u^{\star}_{\mathrm{UC}}(k+1)$ inside the feasible region, the constrained optimal solution can be obtained based on the unconstrained optimal solution as (SILVA; FLESCH; NORMEY-RICO, 2020)

$$\Delta u^{\star}(k) = \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k), U_{g_{\min}}, U_{g_{\max}}), \tag{5.28}$$

where

$$\begin{aligned} U_{g_{\min}} &= \max(\Delta u_{\min}, u_{\min} - u(k-1)), \\ U_{g_{\max}} &= \min(\Delta u_{\max}, u_{\max} - u(k-1)). \end{aligned} \tag{5.29}$$

If the second increment of control action, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, violates any of the constraints, i.e., if

$$\begin{aligned} &\Delta u^{\star}_{\mathrm{UC}}(k+1) > \min\{\Delta u_{\max}, u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k)\}, \\ &\text{or if} \\ &\Delta u^{\star}_{\mathrm{UC}}(k+1) < \max\{\Delta u_{\min}, u_{\min} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k)\} \end{aligned} \tag{5.30}$$

is true, the analysis made in the previous sections can be used with some modifications, due to the fact that the polytope formed by the constraints can vary its form at each sampling instant. The following generalized case is presented in order to show the development of the analytical solution.

Consider that at a given instant k the computed increment of control action, $\Delta\mathbf{u}^{\star}_{\mathrm{UC}}$, is in a certain region of the $\Delta u(k) \times \Delta u(k+1)$ plane, as shown in Figure 35.

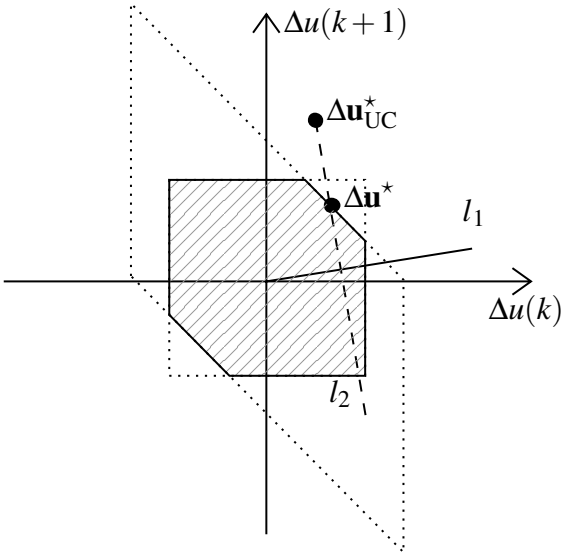


Figure 35 – Projection of the unconstrained optimal solution onto the polytope for the case with constraints both in increment and magnitude for $N_u = 2$

For the particular case with $N_u = 2$ and considering constraints in both increment and magnitude of the control action simultaneously, there are only two possible vectors which define the direction of line l_1: $\mathbf{v_1} = [v_{a_1}, v_{b_1}]$ and $\mathbf{v_2} = [v_{a_2}, v_{b_2}]$. Both vectors can be computed using Equation (5.8), where $\mathbf{v_1}$ is calculated considering $i = 2$ (or $i = 4$) and $\mathbf{v_2}$ considering $i = 6$ (or $i = 8$). Based on this statement, there are only two possible angular coefficients for line l_2: γ_1 and γ_2. These coefficients can be computed as

$$\begin{aligned} \gamma_1 &= -\frac{v_{a_1}}{v_{b_1}}, \\ \gamma_2 &= -\frac{v_{a_2}}{v_{b_2}}. \end{aligned} \tag{5.31}$$

The computation of the term d_x, must be done based on the main active constraint at the sampling instant. For that purpose, in addition to the verification presented in Equation (5.30), it is necessary to verify if the condition $\Delta u_{\max} < u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k)$ is satisfied; in such a case, the increment constraint must be chosen as the main active constraint; otherwise, the magnitude constraint must be chosen as the main active constraint. In case the increment constraint is the main active constraint, d_x is obtained from Equation (5.18) considering $\gamma = \gamma_1$; otherwise Equation (5.21) must be used considering $\gamma = \gamma_2$. From these two cases, it is possible to define a general solution for d_x, using functions $\min(\cdot)$ and $\max(\cdot)$ as

$$d_x = \frac{\Delta u^{\star}_{\mathrm{UC}}(k+1) - \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k+1), U_{f_{\min}}, U_{f_{\max}})}{K_f}, \tag{5.32}$$

where

$$\begin{aligned} U_{f_{\min}} &= \max(\Delta u_{\min}, u_{\min} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k)), \\ U_{f_{\max}} &= \min(\Delta u_{\max}, u_{\max} - u(k-1) - \Delta u^{\star}_{\mathrm{UC}}(k)), \\ K_f &= \begin{cases} K_{\Delta u}, & \text{if } \Delta u_{\max} > U_{1_{\max}} \text{ or } \Delta u_{\min} < U_{1_{\min}} \\ K_u, & \text{else.} \end{cases} \end{aligned} \tag{5.33}$$

and

$$\begin{aligned} K_{\Delta u} &= -\gamma_1, \\ K_u &= -\gamma_2 - 1. \end{aligned} \tag{5.34}$$

Finally, the constrained optimal solution, $\Delta u^{\star}(k)$, can be computed as

$$\Delta u^{\star}(k) = \mathrm{sat}(\Delta u^{\star}_{\mathrm{UC}}(k) + d_x, U_{g_{\min}}, U_{g_{\max}}). \tag{5.35}$$

As all the parameters used to compute γ_1, γ_2, $K_{\Delta u}$, and K_u are available a priori, all these coefficients can be obtained offline leading to a faster computation of the optimal constrained solution. In addition, this case can also be represented using the same block diagram shown in Figure 33. In this case, $\mathrm{sat}_1(\cdot)$ has lower and upper saturation limits of $U_{f\min}$ and $U_{f\max}$, respectively, $\mathrm{sat}_2(\cdot)$ has lower and upper saturation limits of $U_{g_{\min}}$ and $U_{g_{\max}}$, respectively, and K_f is obtained using (5.33) and (5.34).

The above formulation can also be presented in a pseudo-code form, as shown in Algorithm 7.

5.3 PI/PID BASED ON GPC FOR SHORT CONTROL HORIZONS

In this section, a PI/PID design based on GPC is presented. The formulation consists in representing the GPC scheme considering a control horizon of $N_u = 2$ as a PI, or PID controller, plus a control action predictor which, used together with the CSG approach presented in Section 5.2, is capable of computing an optimal control action for first or second-order processes with constraints both in increment and magnitude of control action.

5.3.1 2DOF GPC

A different form to compute the GPC control law can be formulated based on the vector form of the future output predictions, which is given by (NORMEY-RICO; CAMACHO, 2007)

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{F}\mathbf{u_1} + \mathbf{S}\mathbf{y_1}, \tag{5.36}$$

where $\hat{\mathbf{y}} = [\hat{y}(k+1|k), \hat{y}(k+2|k), \ldots, \hat{y}(k+N|k)]^T$, $\mathbf{u_1} = [\Delta u(k-1), \Delta u(k-2), \ldots, \Delta u(k-n_b)]^T$, $\mathbf{y_1} = [y(k), y(k-1), \ldots, y(k-n_a)]^T$, $\mathbf{F} \in \mathbb{R}^{N \times n_b}$ and $\mathbf{S} \in \mathbb{R}^{N \times n_a+1}$ are constant matrices[1], and n_a and n_b are the degrees of the plant model polynomials $A(z^{-1})$ and $B(z^{-1})$, respectively.

[1] The computation of the elements of **F** and **S** is shown in detail in the Appendix A of this work

Algorithm 7: Fast computing of the optimal solution considering constraints in increment and magnitude of control action

```
initialize variables;
compute γ1 and γ2 as in (5.31);
compute K_Δu and K_u as in (5.34) ;
repeat
    compute Δu*_UC as in (2.38);
    compute U_fmin and U_fmax as in (5.33);
    compute U_gmin and U_gmax as in (5.29);
    if Δu*_UC(k+1) > U_fmax then
        if Δu_max < u_max − u(k−1) − Δu*_UC(k) then
            K_f ← K_Δu;
        else
            K_f ← K_u;
        compute d_x as in (5.32);
    else if Δu*_UC(k+1) < U_fmin then
        if Δu_min > u_min − u(k−1) − Δu*_UC(k) then
            K_f ← K_Δu;
        else
            K_f ← K_u;
        compute d_x as in (5.32);
    else
        d_x = 0;
    compute Δu*(k) as in (5.35);
    u*(k) ← u(k−1) + Δu*(k);
    apply u*(k) to the plant;
    update variables;
    k ← k+1;
    wait T_s;
until controller is stopped;
```

Substituting Equation (5.36) in Equation (2.26) and minimizing J_{MPC} with respect to $\Delta\mathbf{u}$ results in

$$\Delta\mathbf{u}^{\star}_{\text{UC}} = \mathbf{M}^{-1}\mathbf{P_0}\mathbf{y_1} + \mathbf{M}^{-1}\mathbf{P_1}\mathbf{u_1} + \mathbf{M}^{-1}\mathbf{G}^{T}\mathbf{r}, \tag{5.37}$$

where $\mathbf{M} = \mathbf{G^T G} + \lambda\mathbf{I} \in \mathbb{R}^{N\times N}$, $\mathbf{P_0} = -\mathbf{G}^{T}\mathbf{S} \in \mathbb{R}^{N\times(n_a+1)}$, $\mathbf{P_1} = -\mathbf{G}^{T}\mathbf{F} \in \mathbb{R}^{N\times n_b}$, and $\mathbf{P_2} = \mathbf{G}^{T} \in \mathbb{R}^{N\times N}$.

Thus, considering $\mathbf{m}$ as the first row of $\mathbf{M}^{-1}$ and the future reference signals $r(k+j) = r(k)$, for $j \geq 1$, it is possible to find $\Delta u^{\star}_{\text{UC}}(k)$ using

$$\Delta u^{\star}_{\text{UC}}(k) = l_{y_1}y(k) + \ldots + l_{y_{n_a+1}}y(k-n_a) + l_{u_1}\Delta u(k-1) + \ldots + l_{u_{n_b}}\Delta u(k-n_b) + \sum_{i=1}^{N} v_i r(k). \tag{5.38}$$

In Equation (5.38), the coefficients $[l_{y_1},\ldots,l_{y_{n_a+1}}]$, $[l_{u_1},\ldots,l_{u_{n_b}}]$, and $[v_1,\ldots,v_N]$ can be obtained

as

$$\begin{aligned}\mathbf{mP_0} &= [l_{y_1}, \ldots, l_{y_{n_a+1}}],\\ \mathbf{mP_1} &= [l_{u_1}, \ldots, l_{u_{n_b}}],\\ \mathbf{mP_2} &= [v_1, \ldots, v_N].\end{aligned} \tag{5.39}$$

The incremental control law presented in Equation (5.38) can be represented as a classical 2DOF structure, with a reference filter, $F_{\text{GPC}}(z^{-1})$, and a feedback controller, $C_{\text{GPC}}(z^{-1})$, which are given by

$$F_{\text{GPC}}(z^{-1}) = \frac{-\sum_{i=1}^{N} v_i}{l_{y_1} + l_{y_2} z^{-1} + \ldots + l_{y_{n_a+1}} z^{-n_a}}, \tag{5.40}$$

$$C_{\text{GPC}}(z^{-1}) = \frac{-(l_{y_1} + l_{y_2} z^{-1} + \ldots + l_{y_{n_a+1}} z^{-n_a})}{(1-z^{-1})(1 - l_{u_1} z^{-1} - \ldots - l_{u_{n_b}} z^{-n_b})}. \tag{5.41}$$

5.3.2 PI/PID tuning based on GPC

Consider a first-order linear process in the discrete-time domain given by

$$P(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0}{1 + a_1 z^{-1}} z^{-1}. \tag{5.42}$$

In this case, the degrees of polynomials $A(z^{-1})$ and $B(z^{-1})$ are, respectively, $n_a = 1$ and $n_b = 0$. In such a case, the incremental control law presented in Equation (5.38) can be represented as a 2DOF PI controller in the discrete-time domain, $C_{\text{PI}}(z^{-1})$, with a reference filter, $F_{\text{PI}}(z^{-1})$, given by

$$C_{\text{PI}}(z^{-1}) = K_p + K_i T_s \frac{1}{1 - z^{-1}}, \tag{5.43}$$

$$F_{\text{PI}}(z^{-1}) = \frac{-\sum_{i=1}^{N} v_i}{l_{y_1} + l_{y_2} z^{-1}}, \tag{5.44}$$

where the proportional and integral gains, K_p and K_i can be computed as

$$\begin{aligned}K_p &= l_{y_2},\\ K_i &= \frac{-l_{y_1} - l_{y_2}}{T_s}.\end{aligned} \tag{5.45}$$

For a second-order linear process given by

$$P(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-1}, \tag{5.46}$$

the degrees of polynomials $A(z^{-1})$ and $B(z^{-1})$ are, respectively, $n_a = 2$ and $n_b = 1$. In this case, the incremental control law presented in Equation (5.38) can be represented as a 2DOF PID controller in the discrete-time domain, $C_{\text{PID}}(z^{-1})$, with a reference filter, $F_{\text{PID}}(z^{-1})$, given by

$$\begin{aligned}C_{\text{PID}}(z^{-1}) &= K_p + K_i T_s \frac{1}{1 - z^{-1}} + \frac{K_d}{\alpha + T_s \frac{1}{1 - z^{-1}}},\\ F_{\text{PID}}(z^{-1}) &= \frac{-\sum_{i=1}^{N} v_i}{l_{y_1} + l_{y_2} z^{-1} + l_{y_3} z^{-2}},\end{aligned} \tag{5.47}$$

where K_d is the derivative gain and α is the parameter of the filter of the derivative part. The PID parameters K_p, K_i, K_d, and α can be computed as

$$\begin{aligned} K_p &= \frac{l_{y_2} + 2l_{y_3} + l_{u_1}(l_{y_1} - l_{y_3})}{(l_{u_1} - 1)^2}, \\ K_i &= \frac{l_{y_1} + l_{y_2} + l_{y_3}}{T_s(l_{u_1} - 1)}, \\ K_d &= \frac{T_s(l_{u_1}^2 l_{y_1} + l_{u_1} l_{y_2} + l_{y_3})}{(l_{u_1} - 1)^3}, \\ \alpha &= \frac{l_{u_1} T_s}{1 - l_{u_1}}. \end{aligned} \tag{5.48}$$

5.3.3 PI/PID controller with constraints handling

Based on the analytical solution for the constrained GPC, considering a short control horizon of $N_u = 2$, shown in Section 5.2, and in the PI/PID representation, presented in Section 5.3.2, it is possible to formulate a 2DOF PI/PID structure which is capable of handling constraints in increment and magnitude of control action providing the exactly same performance of the GPC.

As presented in Section 5.2, all the proposed approaches to deal with the input constraints make use of the unconstrained increment of control action, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, to compute the optimal control action. Thus, an additional block to compute the extra signal must be added to the 2DOF structure.

The extra signal, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, can be obtained based on (5.37). Considering $\tilde{\mathbf{m}}$ as the second row of $\mathbf{M}^{-1}$, $\Delta u^{\star}_{\mathrm{UC}}(k+1)$ can be computed as

$$\begin{aligned} \Delta u^{\star}_{\mathrm{UC}}(k+1) = &\tilde{l}_{y_1} y(k) + \ldots + \tilde{l}_{y_{n_a+1}} y(k-n_a) + \\ &\tilde{l}_{u_1} \Delta u(k-1) + \ldots + \tilde{l}_{u_{n_b}} \Delta u(k-n_b) + \\ &\sum_{i=1}^{N} \tilde{v}_i r(k), \end{aligned} \tag{5.49}$$

where

$$\begin{aligned} \tilde{\mathbf{m}}\mathbf{P_0} &= [\tilde{l}_{y_1}, \ldots, \tilde{l}_{y_{n_a+1}}], \\ \tilde{\mathbf{m}}\mathbf{P_1} &= [\tilde{l}_{u_1}, \ldots, \tilde{l}_{u_{n_b}}], \\ \tilde{\mathbf{m}}\mathbf{P_2} &= [\tilde{v}_1, \ldots, \tilde{v}_N]. \end{aligned} \tag{5.50}$$

Thus, based on the block diagram shown in Figure 33, it is possible to define a structure with a PI/PID controller with the CSG approach which is capable to handle constraints in increment and magnitude of control action. The structure presented in Figure 36 uses an extra block, named predictor (described in Figure 37), which computes the unconstrained future control action $\Delta u^{\star}_{\mathrm{UC}}(k+1)$, based on Equation (5.49), and sends this signal to the CSG block (which has the same structure as the one shown in Figure 33). The CSG block is responsible for

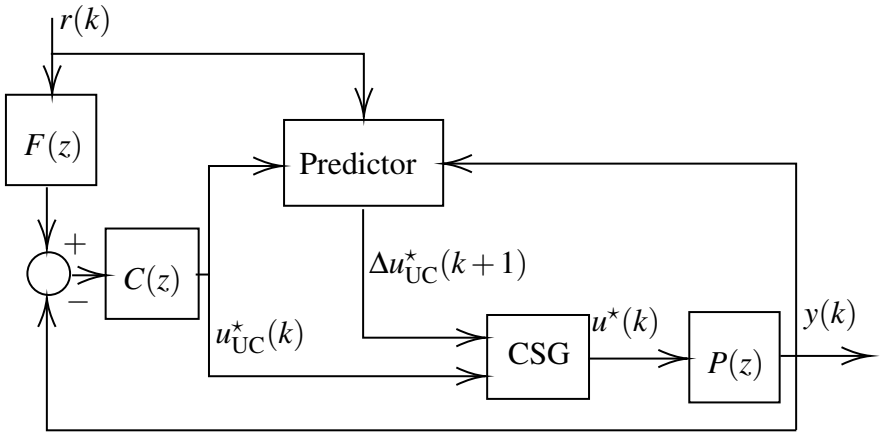


Figure 36 – PI/PID structure with input constraints handling

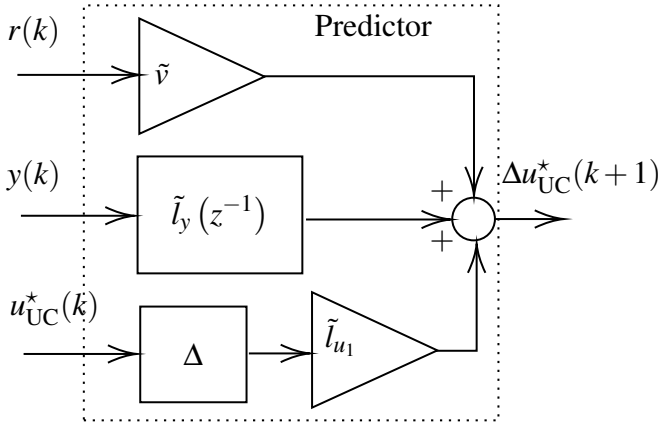


Figure 37 – Future control action predictor

correcting the unconstrained optimal control action in order to satisfy all the constraints. In Figure 37, $\tilde{v}$ and $\tilde{l}_y(z^{-1})$ are given by

$$\begin{aligned} \tilde{v} &= \sum_{i=1}^{N} \tilde{v}_i, \\ \tilde{l}_y(z^{-1}) &= \tilde{l}_{y_1} + \tilde{l}_{y_2} z^{-1} + \ldots + \tilde{l}_{y_{n_a+1}} z^{-n_a}, \end{aligned} \tag{5.51}$$

and the gain $\tilde{l}_{u_1}$ only exists when the equivalent controller is a PID, i.e. when a second-order plant is considered, otherwise $\tilde{l}_{u_1} = 0$.

In order to facilitate the usability of the proposed 2DOF PI/PID with CSG structure, a script implemented in MATLAB®, which computes the PI/PID controller, the reference filter parameters, the predictor, and the $K_{\Delta u}$ and K_u coefficients, based on the GPC tuning parameters (N, N_u, and λ) and also based on the type of constraints considered (increment, magnitude or increment and magnitude of control action simultaneously) is available at https://rodolfoflesch.prof.ufsc.br/pid-from-gpc/.

5.4 CASE STUDY

In order to evaluate the proposed approach, a case study consisting of the control of a second-order underdamped process, used to describe the behavior of a DC motor in Khan et al. (2008), given by

$$P(s) = \frac{2}{s^2 + 12s + 24}, \tag{5.52}$$

is considered. The discrete-time representation for the process with a zero-order hold and sampling time of $T_s = 0.02\,\text{s}$ is

$$P(z) = \frac{0.00037(z + 0.9231)}{z^2 - 1.778z + 0.786}. \tag{5.53}$$

The PID controller based on the GPC was tuned to obtain a fast response. The tuning parameters were set as $N = 10$, $N_u = 2$, and $\lambda = 0$. The simulation of the closed-loop system considers constraints in both increment and magnitude of control action, as $\Delta u_{\min} = -1$, $\Delta u_{\max} = +1$, $u_{\min} = 0$, and $u_{\max} = +14$.

The PID controller and reference filter obtained based on the process tuning parameters are

$$\begin{aligned} C(z^{-1}) &= \frac{4085(1 - 1.256z^{-1} + 0.419z^{-2})}{(1 - z^{-1})(1 + 0.742z^{-1})}, \\ F(z^{-1}) &= \frac{0.1633z^2}{z^2 - 1.256z + 0.419}. \end{aligned} \tag{5.54}$$

As the problem considers constraints both in increment and magnitude of control action, the matrices that describe these constraints can be written as equations (2.41) and (2.43), with $\mathbf{I} \in \mathbb{R}^{2\times 2}$, $\mathbf{T} \in \mathbb{R}^{2\times 2}$, $\mathbf{1} \in \mathbb{R}^2$ and without considering the terms with respect to output constraints.

Firstly, to compute the CSG parameters, vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ must be obtained using Equation (5.8), where $\mathbf{v_1}$ is computed considering $i = 2$ (or $i = 4$) and $\mathbf{v_2}$ is computed considering $i = 6$ (or $i = 8$). The obtained values for these vectors are

$$\mathbf{v_1} = [1, 0.819], \quad \mathbf{v_2} = [4.884, 3.884]. \tag{5.55}$$

Based on $\mathbf{v_1}$ and $\mathbf{v_2}$, it is possible to compute the values of γ_1 and γ_2 using Equation (5.31), resulting in

$$\gamma_1 = 1.2210, \quad \gamma_2 = 1.2575. \tag{5.56}$$

Finally, the gains $K_{\Delta u}$ and K_u can be computed using Equation (5.34), resulting in

$$K_{\Delta u} = 1.221, \quad K_u = 0.2575. \tag{5.57}$$

The predictor parameters can be obtained using equations (5.50) and (5.51), resulting in

$$\begin{aligned} &\tilde{v} = -190.09, \\ &\tilde{l}_y(z^{-1}) = 1257 - 1410z^{-1} + 343.2z^{-2}, \\ &\tilde{l}_{u_1} = 0.2656. \end{aligned} \tag{5.58}$$

Figure 38 shows the simulation of the closed-loop system and compares the 2DOF-PID based on GPC considering four cases: with no AW; with input clipping technique, which consists in clipping the unconstrained control action at the saturation levels (KURTZ; HENSON, 1997); with back-calculation (BC) technique with a tracking time constant of $T_t = 1\,\text{s}$; and with the proposed CSG structure. The simulation considers a step reference of amplitude of 0.5 at $t = 0.5\,\text{s}$ and a load disturbance step of amplitude 5 at $t = 1.5\,\text{s}$.

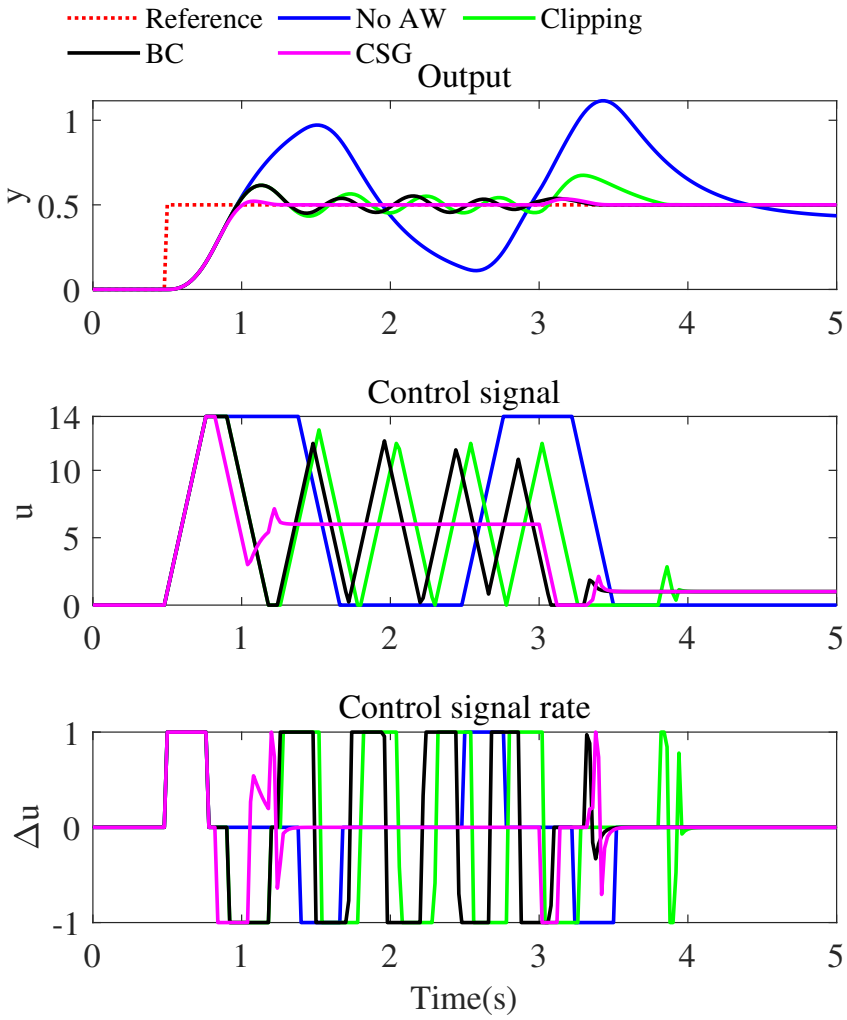


Figure 38 – Performance comparison between the 2DOF PID based on the GPC considering no anti-windup, clipping technique, back-calculation technique (BC), and the proposed approach (CSG)

As can be seen, in the case where no anti-windup technique was considered, the process output exhibited an oscillatory behavior. The clipping technique also presented an oscillatory behavior for reference tracking and a slow disturbance rejection. The back-calculation technique also presented an oscillatory behavior for reference tracking, but was able to reject the disturbance considerably fast. On the other hand, the proposed CSG approach, which has the exactly same response as the constrained GPC, presented good reference tracking and disturbance rejection performance with no oscillations. It is also important to notice that the CSG approach presented the lowest control effort when compared to the other techniques.

Another analysis based on execution time to compute the optimal solution was performed in each sampling interval considering the constrained GPC, using the solver *quadprog* to find the optimal solution, and the PID with the CSG approach was performed. The simulations were performed on a computer with an AMD Ryzen 5 processor with 3.60 GHz of clock, and 16 GB of RAM. The runtimes were measured based on the *tic-toc* command of MATLAB®. The optimality tolerance parameter of the solver *quadprog*, considering the GPC method, was set to 10^{-6}. Figure 39 shows the execution time to compute the optimal solution considering both approaches. A log-scale was used in the $y-$axis for a better illustration of the results. The results show that the execution time to compute the optimal solution at each sampling interval of the PID with CSG is significantly lower than the time required by the original GPC. This happens due to the fact that the proposed algorithm only uses a few if statements and simple arithmetic operations.

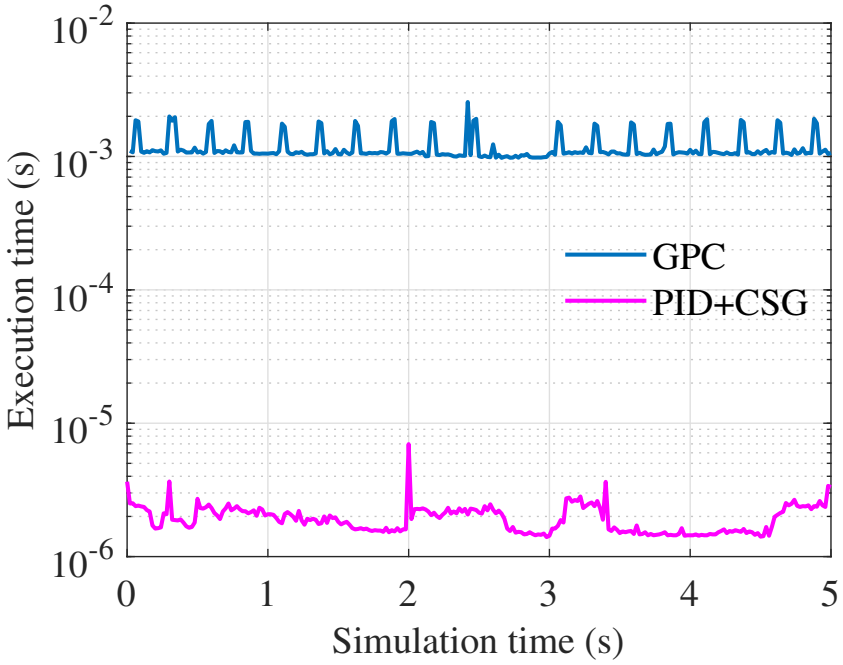


Figure 39 – Execution time comparison of the optimization method between GPC and PID with CSG approach, both with $N = 10$ and $N_u = 2$, for the simulation of the closed-loop system

Table 1 shows the comparative analysis between both approaches, considering mean and maximum (worst case) execution time.

Table 1 – Mean and worst case execution time of both approaches

Method	**Mean**	**Worst case**
GPC (*quadprog*)	1200 µs	2600 µs
PID with CSG	2 µs	7 µs

As can be seen in Table 1, the PID with CSG has a noticeable advantage in terms of mean and worst case execution times when compared to GPC using *quadprog* solver. Furthermore, considering the implementation of the GPC for a practical application, such as in a

microcontroller, the implementation complexity of the proposed method is significantly lower when compared to the implementation of a regular QP solver.

5.5 FINAL CONSIDERATIONS

This chapter presented a 2DOF PI/PID tuning based on the GPC strategy with a governor approach which is capable to handle input constraints and provide exactly the same performance of the constrained GPC without the need to solve an online optimization problem. Firstly, it was shown that considering a short control horizon of $N_u = 2$ it is possible to compute an analytical solution for the constrained optimization problem of the GPC by applying a linear transformation on the unconstrained solution and using the concept of Euclidean distance. Based on this approach, the CSG structure, which is similar to the ones presented in governor strategies, was proposed in order to facilitate the understanding and the usability of the method for dealing with input constraints. Furthermore, it was shown how to compute the gains for a 2DOF PI, or PID, controller in order to obtain the same performance of the unconstrained GPC.

A case study considering a second-order process with input constraints was presented in order to evaluate the proposed method. The presented results show that the 2DOF PID with CSG structure was able to handle all the constraints considered and also presented better reference tracking and disturbance rejection response when compared to other methods typically used in practical applications. Furthermore, a computation time analysis between the original GPC and the 2DOF PID with CSG shows that the mean and worst case execution times to obtain the optimal solution at each sampling interval are significantly lower for the PID with CSG when compared to GPC, mainly due to its simplicity of implementation. This characteristic makes the proposed approach very convenient for simple applications considering fast dynamics and input constraints, where the controller must be implemented in a low-cost hardware.

6 CONCLUSIONS

The research presented in this thesis makes significant contributions to the analysis and design of controllers for systems with characteristics commonly found in industry, such as dead time, constraints in controlled and manipulated variables, modeling errors, and measurement noise. The challenges associated with controlling such processes are discussed, and guidelines, insights, and tools are presented in order to facilitate the selection and implementation of the most suitable controller structures based on the process characteristics. These contributions are presented in the form of three papers.

The first paper, titled "Controlling Industrial Dead-Time Systems: When to Use a PID or an Advanced Controller," compares the performance and robustness of the classic PID controller with advanced control strategies, including DTC and MPC. The comparison considers various case studies, including one experimental case, involving different scenarios, such as dead time, modeling errors, measurement noise, and input and output constraints. The study concludes that the choice of the best control strategy depends more on the quality of the obtained process model than on the magnitude of the dead time. In an industrial context where robust solutions are required, PID control with an AW structure can provide performance as good as or better than more complex strategies such as DTC and MPC. However, when robustness is not a concern and fast performance is desired, the use of DTC or MPC becomes more advantageous, even considering small dead time.

The second paper, titled "CSPS: An Interactive Tool for Control Design and Analysis of Processes with Industrial Characteristics," presents a user-friendly tool for analyzing and simulating systems considering dead time, constraints, measurement noise, modeling errors, and various control strategies and structures. The tool offers several features that enhance its usability, including easy tuning of various controller types and the ability to incorporate AW structures in closed-loop simulations. This feature is particularly beneficial for handling constraints in manipulated and controlled variables. Additionally, the tool provides valuable information on system performance and robustness based on measurement indexes commonly used in industry. The tool also serves as a valuable resource for students and control engineers, aiding in the comprehension of complex concepts related to controlling systems with industrial characteristics. To illustrate the capabilities of the tool, a case study was presented, comparing the performance and robustness of a DTC and an MPC method in the control of an IPDT process subjected to modeling errors and measurement noise.

The third and final paper, titled "PID Design Method Based on GPC with Input Constraints Handling," introduces a novel approach to PID tuning that incorporates the Control Signal Governor (CSG) method for effectively managing input constraints. The methodology is designed to control different types of processes, including stable, integrating, and unstable ones, while achieving optimal performance comparable to the original GPC approach for short control horizons. By integrating the CSG scheme into the PID controller, the proposed method successfully computes the optimal solution for constrained control without the need for an on-

line optimizer. This feature makes it particularly suitable for systems with fast dynamics or limited hardware resources for implementation, such as low-cost microcontrollers.

To evaluate the effectiveness of the approach, a simulation study is conducted on a second-order underdamped process subjected to input constraints. The performance of the proposed method is compared to other control structures considering the same tuning of the controller. The results demonstrate that the proposed approach effectively handles all the constraints and exhibits significantly improved performance, characterized by reduced overshoot and oscillations in the process output, as well as low variability in the control action. Additionally, an analysis comparing the computational execution time required to obtain the control action of the proposed approach and the original GPC approach reveals that the proposed method exhibits considerably faster execution time, in both mean and worst case scenarios, mainly due to its simplicity of implementation.

Overall, the achievements of this thesis align with the stated objectives and also make significant scientific contributions by providing valuable insights and practical tools for control engineers that work with the challenging industrial processes.

In order to improve the contributions of the research presented in this thesis, several suggestions for future works can be considered.

One possible line for future research is to extend the approach for controlling processes with output constraints, as presented in Chapter 3. This extension could specifically focus on cases that involve large control horizons, thereby exploring the applicability and effectiveness of the approach in more complex systems.

Additionally, improvements to the interface of the CSPS tool, which was discussed in Chapter 4, can be made. Those improvements may include enhancing its usability and incorporating additional control strategies and other methods to handle process constraints. By expanding the tool capabilities, researchers and practitioners would have a more comprehensive toolkit for designing and evaluating control strategies for several industrial processes.

Furthermore, it is interesting to validate the proposed PID with CSG structure, as presented in Chapter 5, through experimental case studies. A potential avenue for validation is to consider a process with fast dynamics, allowing for a thorough examination of the structure performance and robustness in challenging real-world scenarios. Also, expanding the PID with CSG structure to accommodate larger control horizons could also be a valuable area for future exploration.

Lastly, introducing a feedforward capability into the PID with CSG structure could significantly enhance the disturbance rejection response. By incorporating this feature, the controller would be equipped to proactively mitigate disturbances, thereby improving overall system performance.

By pursuing these suggested lines of research, future studies can build upon the findings of this thesis and further advance the field of control systems, ultimately leading to more effective and robust control strategies for a wide range of industrial processes.

BIBLIOGRAPHY

ÅSTRÖM, K. J.; HAGGLUND, T. **PID Controllers: Theory, Design and Tuning**. 2. ed. North Carolina, USA: ISA, 1995.

ÅSTRÖM, K. J.; HÄGGLUND, T. The future of PID control. **Control Engineering Practice**, Elsevier, v. 9, p. 1163–1175, 2001.

ÅSTRÖM, K. J.; WITTENMARK, B. **Computer-controlled systems: theory and design**. 3. ed. Upper Saddle River, NJ: Prentice Hall, 2013.

BAGYAVEERESWARAN, V. et al. Performance comparison of next generation controller and MPC in real time for a SISO process with low cost DAQ unit. **Alexandria Engineering Journal**, Elsevier, v. 55, n. 3, p. 2515–2524, 2016.

BRUCIAPAGLIA, A.; APOLÔNIO, R. Uma estratégia de eliminação da sobrecarga da ação integral para controladores PID discretos: Aplicação no controle de velocidade de um motor de corrente contínua. In: AADECA. **II Congresso Latino Americano de Controle Automático**. Buenos Aires, Argentina, 1986. p. 519–524.

CAMACHO, E. F.; BORDONS, C. **Model Predictive Control**. London: Springer, 2013.

CASTANO, J. A. et al. Enhancing the robustness of the EPSAC predictive control using a singular value decomposition approach. **Robotics and Autonomous Systems**, Elsevier, v. 74, p. 283–295, 2015.

CASTRO, F. A. de et al. Comparison of fractional and integer PID controllers tuned by genetic algorithm. In: **12th IEEE International Conference on Industry Applications (INDUSCON)**. Curitiba, PR, Brazil: IEEE, 2016. p. 1–7.

CAVANINI, L.; CIMINI, G.; IPPOLITI, G. Model predictive control for pre-compensated power converters: Application to current mode control. **Journal of the Franklin Institute**, Elsevier, v. 356, n. 4, p. 2015–2030, mar 2019.

CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control—Part i. The basic algorithm. **Automatica**, Elsevier, v. 23, p. 137–148, 1987.

CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control: A computer control algorithm. In: **Joint Automatic Control Conference**. San Francisco, California: IEEE, 1980. p. 72.

da COSTA FILHO, M. V. A.; NORMEY-RICO, J. E. An interactive tool to design controllers for processes with dead time. **IFAC Proceedings Volumes**, Elsevier, v. 42, n. 14, p. 189–194, 2009.

DOYLE, J. C.; SMITH, R. S.; ENNS, D. F. Control of plants with input saturation nonlinearities. In: IEEE. **American Control Conference, 1987**. Minneapolis, USA: IEEE, 1987. p. 1034–1039.

ELLIS, G. **Control system design guide: using your computer to understand and diagnose feedback controllers**. Waltham, USA: Elsevier, 2012.

FERTIK, H. A.; ROSS, C. W. Direct digital control algorithm with anti-windup feature. **ISA Transactions**, Elsevier, v. 6, n. 4, p. 317, 1967.

FLESCH, R. C. C.; NORMEY-RICO, J. E.; FLESCH, C. A. A unified anti-windup strategy for SISO discrete dead-time compensators. **Control Engineering Practice**, v. 69, p. 50–60, 2017.

FLESCH, R. C. C.; SANTOS, T. L. M.; NORMEY-RICO, J. E. Unified approach for minimal output dead time compensation in MIMO non-square processes. In: **51st IEEE Conference on Decision and Control (CDC)**. Maui, USA: IEEE, 2012. p. 2376–2381.

FORBES, M. G. et al. Model predictive control in industry: Challenges and opportunities. **IFAC-PapersOnLine**, v. 48, p. 531–538, 2015.

GALEANI, S. et al. A tutorial on modern anti-windup design. **European Journal of Control**, Elsevier, v. 15, n. 3-4, p. 418–440, 2009.

GARCIA, C. E.; MORSHEDI, A. Quadratic programming solution of dynamic matrix control (QDMC). **Chemical Engineering Communications**, Taylor & Francis, v. 46, n. 1-3, p. 73–87, 1986.

GARPINGER, O.; HÄGGLUND, T. Software-based optimal PID design with robustness and noise sensitivity constraints. **Journal of Process Control**, Elsevier BV, v. 33, p. 90–101, 2015.

GILBERT, E. G.; TAN, K. T. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. **IEEE Transactions on Automatic Control**, IEEE, v. 36, n. 9, p. 1008–1020, 1991.

GOODWIN-SIN. **Adaptive filtering prediction and control**. New Jersey, USA: Prentice Hall, 1984.

GUDIN, R.; MIRKIN, L. On the delay margin of dead-time compensators. **International Journal of Control**, Taylor & Francis, v. 80, p. 1316–1332, 2007.

GUZMÁN, J. L. et al. Interactive learning modules for PID control. **IFAC Proceedings Volumes**, Elsevier, v. 39, n. 6, p. 7–12, 2006.

GUZMAN, J. L.; BERENGUEL, M.; DORMIDO, S. Interactive teaching of constrained generalized predictive control. **IEEE Control Systems Magazine**, IEEE, v. 25, n. 2, p. 52–66, 2005.

HABER, R.; BARS, R.; SCHMITZ, U. **Predictive Control in Process Engineering-From the Basics to the Applications**. Weinheim, Germany: John Wiley & Sons, 2012.

HALVGAARD, R. et al. Electric vehicle charge planning using economic model predictive control. In: IEEE. **International Electric Vehicle Conference**. Greenville, USA, 2012. p. 1–6.

HIPPE, P. **Windup in Control: Its Effects and Their Prevention**. London: Springer Science & Business Media, 2006.

HOROWITZ, I. Some properties of delayed controls (Smith regulator). **International Journal of Control**, Taylor and Francis Group, v. 38, p. 977–990, 11 1983.

IPOUM-NGOME, P. G. et al. Optimal finite state predictive direct torque control without weighting factors for motor drive applications. **IET Power Electronics**, IET, v. 12, n. 6, p. 1434–1444, 2019.

JEROME, N.; RAY, W. H. High-performance multivariable control strategies for systems having time delays. **American Institute of Chemical Engineers Journal**, Wiley Online Library, v. 32, n. 6, p. 914–931, 1986.

KAPASOURIS, P.; ATHANS, M.; STEIN, G. Design of feedback control systems for stable plants with saturating actuators. In: **Proceedings of the 27th IEEE Conference on Decision and Control**. Austin, Texas: IEEE, 1988. p. 469–479.

KAYA, I. A new Smith predictor and controller for control of processes with long dead time. **ISA Transactions**, v. 42, p. 101 – 110, 2003.

KEYSER, R. D.; IONESCU, C. M. The disturbance model in model based predictive control. In: **IEEE Conference on Control Applications**. Istanbul, Turkey: IEEE, 2003. p. 446–451.

KHAN, S. et al. Design and implementation of an optimal fuzzy logic controller using genetic algorithm. Journal of Computer Science, 2008.

KOTHARE, M. V. et al. A unified framework for the study of anti-windup designs. **Automatica**, Elsevier, v. 30, n. 12, p. 1869–1883, 1994.

KUO, S. M.; MORGAN, D. R. **Active noise control systems**. Toronto: Wiley-Interscience, 1996. v. 4.

KURTZ, M. J.; HENSON, M. A. Input-output linearizing control of constrained nonlinear processes. **Journal of Process Control**, Elsevier, v. 7, n. 1, p. 3–17, 1997.

KWON, W. H.; BYUN, D. G. Receding horizon tracking control as a predictive control and its stability properties. **International Journal of Control**, Taylor & Francis, v. 50, n. 5, p. 1807–1824, 1989.

LIU, T. et al. Heating-up control with delay-free output prediction for industrial jacketed reactors based on step response identification. **ISA Transactions**, Elsevier, v. 83, p. 227–238, 2018.

MAJHI, S.; ATHERTON, D. P. A new Smith predictor and controller for unstable and integrating processes with time delay. In: IEEE. **Proceedings of the 37th IEEE Conference on Decision and Control**. Tampa, USA, 1998. v. 2, p. 1341–1345.

MARQUIS, P.; BROUSTAIL, J. SMOC, a bridge between state space and model predictive controllers: application to the automation of a hydrotreating unit. **IFAC Proceedings Volumes**, Elsevier, v. 21, n. 4, p. 37–45, 1988.

MBUNGU, T. et al. Smart SISO-MPC based energy management system for commercial buildings: Technology trends. In: IEEE. **2016 Future Technologies Conference (FTC)**. San Francisco, USA, 2016. p. 750–753.

MÉNDEZ, J. A. et al. A web-based tool for control engineering teaching. **Computer Applications in Engineering Education**, Wiley Online Library, v. 14, n. 3, p. 178–187, 2006.

MEYER, C. D. **Matrix analysis and applied linear algebra**. Philadelphia: Society for Industrial and Applied Mathematics, 2000.

MILLER, R. et al. Predictive PID. **ISA transactions**, Elsevier, v. 38, n. 1, p. 11–23, 1999.

NORMEY-RICO, J.; BORDONS, C.; CAMACHO, E. Improving the robustness of dead-time compensating PI controllers. **Control Engineering Practice**, Elsevier, v. 5, p. 801–810, 1997.

NORMEY-RICO, J. E.; CAMACHO, E. F. **Control of Dead-time Processes**. London: Springer, 2007.

NORMEY-RICO, J. E.; CAMACHO, E. F. Unified approach for robust dead-time compensator design. **Journal of Process Control**, Elsevier, v. 19, n. 1, p. 38–47, 2009.

NORMEY-RICO, J. E.; GUZMÁN, J. L. Unified PID tuning approach for stable, integrative, and unstable dead-time processes. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 52, p. 16811–16819, 2013.

NORMEY-RICO, J. E. et al. An unified approach for dtc design using interactive tools. **Control Engineering Practice**, Elsevier, v. 17, n. 10, p. 1234–1244, 2009.

O'DWYER, A. **Handbook of PI and PID controller tuning rules**. London: Imperial College Press, 2009.

OSMAN, K. bin et al. GPC controller design for an intelligent pneumatic actuator. **Procedia Engineering**, Elsevier, v. 41, p. 657–663, 2012.

OVIEDO, J. J. E.; BOELEN, T.; OVERSCHEE, P. V. Robust advanced PID control (RaPID): PID tuning based on engineering specifications. **IEEE Control Systems**, IEEE, v. 26, n. 1, p. 15–19, 2006.

PALMOR, Z. Stability properties of Smith dead-time compensator controllers. **International Journal of Control**, Taylor & Francis, v. 32, p. 937–949, 1980.

PANDA, R. C.; HUNG, S.-B.; YU, C.-C. An integrated modified smith predictor with pid controller for integrator plus deadtime processes. **Industrial & engineering chemistry research**, ACS Publications, v. 45, n. 4, p. 1397–1407, 2006.

PANDA, R. C.; YU, C.-C.; HUANG, H.-P. PID tuning rules for SOPDT systems: Review and some new results. **ISA Transactions**, Elsevier, v. 43, p. 283–295, 2004.

PENG, Y.; VRANCIC, D.; HANUS, R. Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. **Control Systems, IEEE**, IEEE, v. 16, n. 4, p. 48–57, 1996.

QIN, S. J.; BADGWELL, T. A. A survey of industrial Model Predictive Control Technology. **Control Engineering Practice**, Elsevier, v. 11, n. 7, p. 733–764, 2003.

RIVERA, D. E.; MORARI, M.; SKOGESTAD, S. Internal model control: PID controller design. **Industrial & Engineering Chemistry Process Design and Development**, ACS Publications, v. 25, p. 252–265, 1986.

RODRIGUES, M.; ODLOAK, D. Output feedback MPC with guaranteed robust stability. **Journal of Process Control**, Elsevier, v. 10, n. 6, p. 557–572, 2000.

RUNDQWIST, L.; STÅHL-GUNNARSSON, K.; ENHAGEN, J. Rate limiters with phase compensation in jas 39 gripen. In: IEEE. **1997 European Control Conference (ECC)**. Brussels, Belgium, 1997. p. 3944–3949.

SALEM, F.; MOSAAD, M. I. A comparison between MPC and optimal PID controllers: Case studies. In: **Michael Faraday IET International Summit**. Kolkata, India: Institution of Engineering and Technology, 2015. p. 59–65.

SANZ, R.; GARCÍA, P.; ALBERTOS, P. A generalized Smith predictor for unstable time-delay SISO systems. **ISA transactions**, Elsevier, v. 72, p. 197–204, 2018.

SATO, T. Design of a GPC-based PID controller for controlling a weigh feeder. **Control Engineering Practice**, Elsevier, v. 18, n. 2, p. 105–113, 2010.

SEBORG, D.; EDGAR, T. F.; MELLICHAMP, D. **Process dynamics & control**. New Jersey, USA: John Wiley & Sons, 2006.

SERON, M. M.; DONA, J. A. D.; GOODWIN, G. C. Global analytical model predictive control with input constraints. In: IEEE. **Proceedings of the 39th IEEE Conference on Decision and Control**. Sydney, NSW, Australia, 2000. v. 1, p. 154–159.

SHA'ABAN, Y. A.; LENNOX, B.; LAURÍ, D. PID versus MPC performance for SISO dead-time dominant processes. In: **10th IFAC International Symposium on Dynamics and Control of Process Systems**. Mumbai, India: IFAC Proceedings Volumes, 2013. p. 241–246.

SILVA, L. R. da; FLESCH, R. C. C.; NORMEY-RICO, J. E. Analysis of anti-windup techniques in PID control of processes with measurement noise. In: **3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control**. Ghent, Belgium: IFAC-PapersOnLine, 2018. p. 948–953.

SILVA, L. R. da; FLESCH, R. C. C.; NORMEY-RICO, J. E. Controlling industrial dead-time systems: When to use a PID or an advanced controller. **ISA Transactions**, Elsevier, v. 99, p. 339–350, 2020.

SKOGESTAD, C. G. S. Should we forget the smith predictor? **IFAC-PapersOnLine**, Elsevier, v. 51, n. 4, p. 769–774, 2018.

SMITH, J. O. Closed control of loops with dead time. **Chemical Engineering Progress**, v. 53, p. 217–219, 1957.

SOETERBOEK, R. **Predictive control: a unified approach**. New York, USA: Prentice-Hall, 1992.

STEIN, G. Respect the unstable. **IEEE Control systems magazine**, IEEE, v. 23, n. 4, p. 12–25, 2003.

SUDIBYO et al. Comparison of MIMO MPC and PI decoupling in controlling methyl tert-butyl ether process. **Computer Aided Chemical Engineering**, v. 31, p. 345–349, 2012.

TAKAO, K.; YAMAMOTO, T.; HINAMOTO, T. A new GPC-based PID controller using memory-based identification. In: IEEE. **Proceedings of the 47th Midwest Symposium on Circuits and Systems**. Hiroshima, Japan, 2004. v. 3, p. 125–128.

TAN, K.; HUANG, S.; LEE, T. Development of a GPC-based PID controller for unstable systems with dead time. **ISA transactions**, Elsevier, v. 39, p. 57–70, 2000.

TAN, K. K. et al. PID control design based on a GPC approach. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 41, p. 2013–2022, apr 2002.

TARBOURIECH, S.; TURNER, M. Anti-windup design: an overview of some recent advances and open problems. **IET Control Theory & Applications**, IET, v. 3, n. 1, p. 1–19, 2009.

TORRICO, B. C. et al. Simple tuning rules for dead-time compensation of stable, integrative, and unstable first-order dead-time processes. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 52, n. 33, p. 11646–11654, 2013.

UDUEHI, D.; ORDYS, A.; GRIMBLE, M. Predictive PID controller design. In: IEEE. **Proceedings of the 2001 IEEE International Conference on Control Applications**. Mexico City, Mexico, 2001. p. 612–617.

VALENCIA-PALOMO, G.; ROSSITER, J. Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information. **ISA Transactions**, Elsevier, v. 50, n. 1, p. 92–100, 2011.

VAZQUEZ, S. et al. Model predictive control for power converters and drives: Advances and trends. **IEEE Transactions on Industrial Electronics**, IEEE, v. 64, n. 2, p. 935–947, 2017.

VISIOLI, A.; ZHONG, Q. **Control of Integral Processes with Dead Time**. London: Springer, 2010.

WANG, L. **Model predictive control system design and implementation using MATLAB®**. London: Springer Science & Business Media, 2009.

YDSTIE, B. Extended horizon adaptive control. **IFAC Proceedings Volumes**, Elsevier, v. 17, n. 2, p. 911–915, 1984.

ZACCARIAN, L.; TEEL, A. R. **Modern Anti-windup Synthesis: Control Augmentation for Actuator Saturation**. New Jersey, USA: Princeton University Press, 2011.

ZHANG, R. et al. Design of fractional order modeling based extended non-minimal state space mpc for temperature in an industrial electric heating furnace. **Journal of Process Control**, Elsevier, v. 56, p. 13–22, 2017.

APPENDIX A – COMPUTATION OF ELEMENTS OF F AND S IN GPC CONTROL LAW

In equation 5.36, matrices **F** and **S** has the form

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1(n_b+1)} \\ f_{21} & f_{22} & \cdots & f_{2(n_b+1)} \\ \vdots & \vdots & \vdots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{N(n_b+1)} \end{bmatrix}, \mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1(na+1)} \\ s_{21} & s_{22} & \cdots & s_{2(na+1)} \\ \vdots & \vdots & \vdots & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{N(na+1)} \end{bmatrix}, \tag{A.1}$$

where the elements of **F** are computed as

$$\begin{aligned} &f_{1j} = b_{j-1}, \text{ for } j = 1,\ldots,n_b, \\ &f_{ij} = -\sum_{d=1}^{i-1} \tilde{a}_d h_{(i-d)j} + h_{1(i+j-1)}, \text{ for } i = 2\ldots,N \text{ and } j = 1\ldots,n_b, \end{aligned} \tag{A.2}$$

where $f_{ij} = 0$ if $i > N$ or $j > n_b$, $\tilde{a}_i$ are the coefficients of $\tilde{A}(z^{-1}) = (1-z^{-1})A(z^{-1})$ and $\tilde{a}_d = 0$ if $d > na+1$.

The elements of **S** can be computed as

$$\begin{aligned} &s_{1j} = -\tilde{a}_j, \text{ for } j = 1\ldots,na+1, \\ &s_{ij} = \sum_{k=1}^{n_a+1} s_{1k} s_{(i-k)j}, \text{ for } i = 2\ldots,N \text{ and } j = 1,\ldots,na+1, \end{aligned} \tag{A.3}$$

where $s_{ij} = 0$ if $i < 0$ or $j < 0$, $s_{0j} = 1$ and $s_{i0} = 1$.