UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Design of a multivariable control for the liquid cooling system of a 2kW PEM stack.

Master's Thesis

Master's Degree in Industrial Engineering

AUTHOR: Eftekharipour , Hamed

Tutor: Blasco Ferragud, Francesc Xavier

External cotutor: MUSSETTA, MARCO

ACADEMIC YEAR: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

INDUSTRIAL ENGINEERING MASTER THESIS

MULTIVARIABLE CONTROL FOR THE LIQUID COOLING SYSTEM OF A 2KW PEM TYPE BATTERY

AUTHOR: HAMED EFTEKHARIPOUR

SUPERVISOR: XAVIER BLASCO FERRAGUD

SUPERVISOR: Marco Mussetta

Academic year: 2021-22

Acknowledgment

I would like to acknowledge and give my warmest thanks to my supervisor (Xavier Blasco Ferragud) who made this work possible. His guidance and advice carried me through all the stages of writing my project. I would also like to give special thanks to my family as a whole for their continuous support and understanding when undertaking my research and writing my project. Your prayer for me was what sustained me this far. Many thanks to the other Professors who motivated me to study and tireless passion for learning during the thesis. Thanks to Alberto Pajares and Santiago Navarro Gimenez.

Abstract

The main objective of this work is to design, adjust and validate a multivariable control structure for the control of a complex process. The process concerned is the temperature control of a PEM type stack that performs the function of prime mover in a micro-CHP (micro Combined Heat and Power) system. It is a system that must supply electrical and thermal energy for domestic applications. Proper control of the thermal subsystem is essential for optimal operation. The work has to propose several alternatives for multivariable control. These structures must be optimally tuned to satisfy two objectives simultaneously: performance and control effort. The tuning problem is posed as a multiobjective optimization problem. In this particular case, there is uncertainty when working with a linear model when the process is non-linear. There is a part of the process dynamics that is not correctly modeled. A significant decrease in performance may occur when the designed controller is applied to the real process. Then an additional step in the controller validation consists of evaluating robustness for model uncertainties. The tasks to be performed in order to achieve the proposed objectives are:

- To understand the system's operation and to obtain linear models that adequately represent the system at least at one point of operation.
- To understand the multiobjective optimization tools to be used for the approach and resolution of the problem of adjustment of the proposed control structures.
- To define the multiobjective optimization problem for the tuning of multivariable controllers.
- Multiobjective adjustment of the different multivariable control structures to be proposed.
- Analysis of results and evaluation of the robustness for validating the proposed structure in the real process.

keywords: Multivariable control, Multiobjective optimization, Linear and nonlinear model

Resumen

El objetivo principal de este trabajo es diseñar, ajustar y validar una estructura de control multivariable para el control de un proceso complejo. El proceso en cuestión es el control de temperatura de una pila tipo PEM que realiza la función de motor primario en un sistema micro-CHP (micro Combined Heat and Power). Es un sistema que debe suministrar energía eléctrica y térmica para aplicaciones domésticas. El control adecuado del subsistema térmico es fundamental para un funcionamiento óptimo. El trabajo tiene que proponer varias alternativas de control multivariable. Estas estructuras deben ajustarse de manera óptima para satisfacer dos objetivos simultáneamente: rendimiento y esfuerzo de control. El problema de sintonía se plantea como un problema de optimización multiobjetivo. En este caso particular, existe incertidumbre al trabajar con un modelo lineal cuando el proceso es no lineal. Hay una parte de la dinámica del proceso que no está correctamente modelada. Puede ocurrir una disminución significativa en el rendimiento cuando el controlador diseñado se aplica al proceso real. Luego, un paso adicional en la validación del controlador consiste en evaluar la robustez para las incertidumbres del modelo. Las tareas a realizar para lograr los objetivos propuestos son:

- Comprender el funcionamiento del sistema y obtener modelos lineales que representen adecuadamente el sistema al menos en un punto de funcionamiento.
- Comprender las herramientas de optimización multiobjetivo a utilizar para el planteamiento y resolución del problema de ajuste de las estructuras de control propuestas.
- Definir el problema de optimización multiobjetivo para el ajuste de controladores multivariables.
- Ajuste multiobjetivo de las diferentes estructuras de control multivariable a proponer.
- Análisis de resultados y evaluación de la robustez para validar la estructura propuesta en el proceso real.

palabras clave: Control multivariable, Optimización multiobjetivo, Modelo lineal y no lineal

Resum

L'objectiu principal d'aquest treball és dissenyar, ajustar i validar una estructura de control multivariable per al control d'un procés complex. El procés en qüestió és el control de la temperatura d'una pila de tipus PEM que fa la funció de motor principal en un sistema micro-CHP (micro Combined Heat and Power). És un sistema que ha de subministrar energia elèctrica i tèrmica per a aplicacions domèstiques. El control adequat del subsistema tèrmic és essencial per a un funcionament òptim. El treball ha de proposar diverses alternatives de control multivariable. Aquestes estructures s'han d'ajustar de manera òptima per satisfer dos objectius simultàniament: rendiment i esforç de control. El problema d'ajust es planteja com un problema d'optimització multiobjectiu. En aquest cas particular, hi ha incertesa quan es treballa amb un model lineal quan el procés és no lineal. Hi ha una part de la dinàmica del procés que no està correctament modelada. Es pot produir una disminució significativa del rendiment quan el controlador dissenyat s'aplica al procés real. A continuació, un pas addicional en la validació del controlador consisteix a avaluar la robustesa per a les incerteses del model. Les tasques a realitzar per assolir els objectius proposats són:

- Comprendre el funcionament del sistema i obtenir models lineals que representin adequadament el sistema almenys en un punt de funcionament.
- Conèixer les eines d'optimització multiobjectiu a utilitzar per a l'enfocament i resolució del problema d'ajust de les estructures de control proposades.
- Definir el problema d'optimització multiobjectiu per a la sintonització de controladors multivariables.
- Ajust multiobjectiu de les diferents estructures de control multivariable a proposar.
- Anàlisi de resultats i avaluació de la robustesa per validar l'estructura proposada en el procés real.

paraules clau: control multivariable, optimització multiobjectiu, model lineal i no lineal

Contents

1 Introduction

The current environmental issues (climate change, air pollution, and the depletion of fossil fuel supplies) highlight the necessity of switching to sustainable energy production methods. Due to its great electrical efficiency and minimal CO_2 emissions, PEMFC stacks offer a cost-effective and environmentally friendly choice for several power production applications: Backup mechanisms, micro-CHP, hybrid electric vehicles (HEV), and hybrid renewable energy systems (HRES)[9]

1.1 Micro-CHP system

This section will go through the Micro Combined Heat and Power (Micro-CHP) system in detail, focusing on the stack cooling system. The proton exchange membrane fuel cell stacks (PEMFC) stack, air supply system, hydrogen supply system, electronic load, radiator, hot water tank, stack cooling system, control unit, and computer are the fundamental components of the micro-CHP system.

Figure 1: experimental equipment[9].

A SCADA is used to monitor and control the plant in the latter. The stack, which is fed hydrogen and air, generates both electricity and heat. The electronic load consumes the electrical energy, allowing us to simulate residential power usage (lighting and appliances). The thermal energy is stored in a heat-buffering hot water tank. This tank's water temperature is roughly $55°C$. The radiator simulates the thermal energy requirement (hot water and heating) by extracting some of the heat stored in the hot water tank when it is turned on. The stack temperature must be kept within safe limits for the system to function correctly and for the stack to be safe. According to the stack manufacturer, this temperature must be kept at $65°C$ to achieve maximum electrical efficiency. As a result, this is its set point. To produce a consistent and restricted temperature gradient in the stack, the temperature difference between stack outflow water and stack inflow water($T_{w_{out}} - T_{w_{in}}$) must also be kept within limitations. This is required to prevent thermal stress in the stack and, as a result, deterioration. According to the stack maker, this temperature gradient should ideally be maintained at $5°C$. As a result, the set-point is set to $60°C$[9].

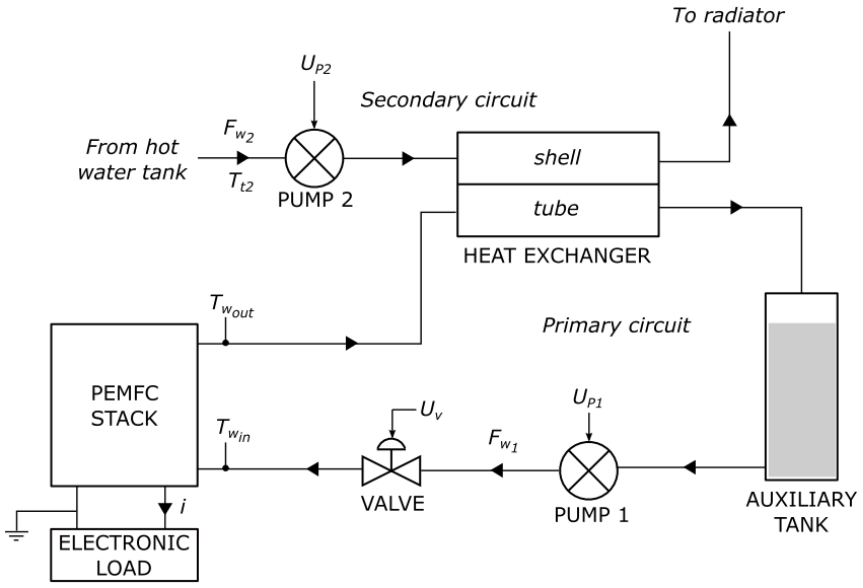


Figure 2: Stack cooling system. The stack temperature $T_{w_{out}}(C)$ is regulated by the water flow rate of the main circuit $F_{w_1}(l/min)$, and the stack inlet water temperature $T_{w_{in}}(C)$ is controlled by the water flow rate of the secondary circuit $F_{w_2}(l/min)$. Unwanted oscillations in $T_{w_{out}}$ will result from changes in the electrical current demand I(A)[9].

The temperature control of the stack is critical for its optimal performance in these applications. This is because the electrical efficiency and longevity of the stack are both dependent on the performance of this control, i.e., a reasonable temperature control boosts the stack's electrical efficiency and lifespan—the stack's electrical efficiency increases as the temperature rises. However, once a particular temperature threshold is surpassed, the electrical efficiency declines due to membrane dryness. Dehydration also accelerates the breakdown of the stack and, as a result, lowers its lifespan.

2 Multi-objective background

This section provides some background information on multi-objective optimization(MOP). MOP, Pareto optimally, Pareto dominance, Pareto optimal set, and Pareto front are all technically defined concepts. Without losing generality, we assume in these formulations that minimization is the goal for all objectives. The following is a formal definition of a generic MOP[3]:

- Definition 1 (MOP): Find a vector $\vec{x^*} = [x_1^*, x_2^*, \cdots, x_n^*]$ that meets the m inequality constraints $g_i(\vec{x}) \geq 0, i = 1, 2, \cdots, m$, the p equality constraints $h_i(\vec{x}) = 0, i = 1, 2, \cdots, p$, and minimizes the vector function $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \cdots, f_k(\vec{x})]^T$, where $\vec{x} = [x_1, x_2, \cdots, x_n]^T$ is the vector of decision variables. The feasible region Ω is defined as the set of all values matching the requirements, and every point $\vec{x} \in \Omega$ is a viable solution. We also seek the Pareto optima.

- Definition 2 (Pareto Optimality): If for every $\vec{x} \in \Omega$ and $I = [1, 2, \cdots, k]$ there is at least one $i \in I$ such that $f_i(\vec{x}) > f_i(\vec{x^*})$, then the point $\vec{x^*} \in \Omega$ is Pareto Optimal. According to this definition, $\vec{x^*}$ is Pareto optimum if no other feasible vector $\vec{x}$ exists that improves certain criteria without deteriorating at least one other criterion.

- Definition 3 (Pareto dominance): If and only if $\vec{u}$ is somewhat less than $\vec{v}$, A vector $\vec{u} = (u_1, \cdots, u_k)$ is said to dominate $\vec{v} = (v_1, \cdots, v_k)$.

- Definition 4 (Pareto optimal set): For a given MOP $\vec{f}(\vec{x})$ The Pareto optimal set is defined as $P^* = [\vec{x} \in \Omega | \exists \vec{x} \in \Omega, \vec{f}(\vec{x}) \leq \vec{f}(\vec{x})]$

- Definition 5 (Pareto front): For a given MOP $\vec{f}(\vec{x})$ and its Pareto optimal set P^*, the Pareto front is defined as $F_p = [\vec{f}(\vec{x}), \vec{x} \in P^*]$

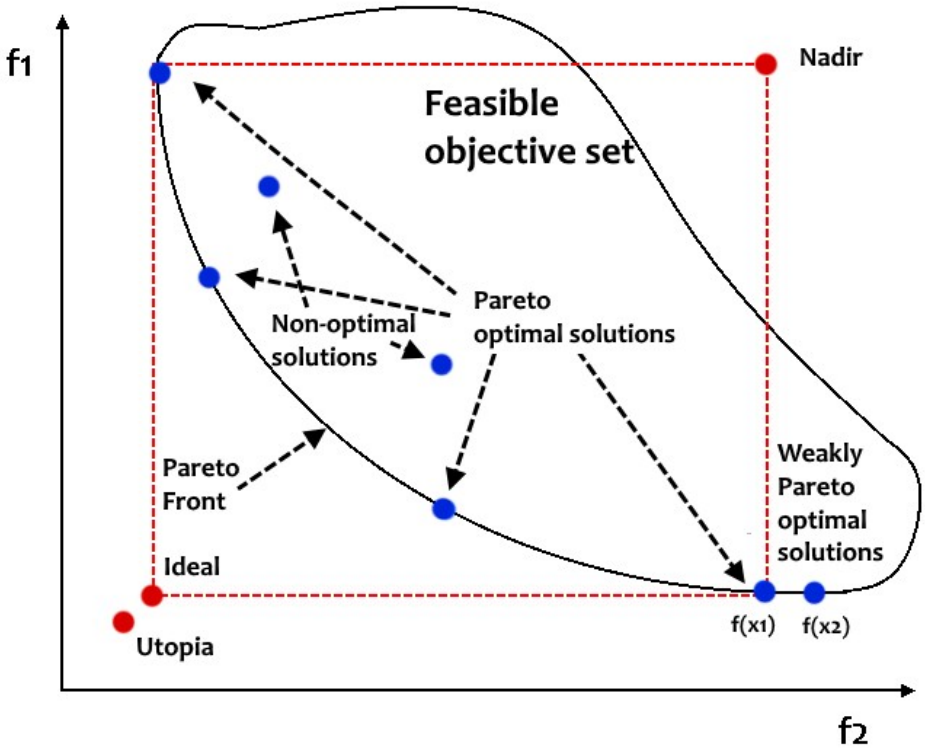


Figure 3: Ideal, nadir and utopia points of two-objective optimization problem, where both objectives are to be minimized[6]

A feasible solution $\acute{x} \in S$ and the accompanying $f_i(\acute{x}) \in Z$ are said to be weakly Pareto optimal if another feasible solution $x \in S$ does not exist such that $f_i(x) \leq f_i(\acute{x})$ for all $i = [1, \cdots, k]$. Furthermore, they are said to be Pareto optimal if no alternative feasible solution $x \in S$ exists such that $f_i(x) \leq f_i(\acute{x})$ for all $i = [1, \cdots, k]$ and$f_j(x) < f_j(\acute{x})$ for at least some $j \in [1, \cdots, k]$. As a result, all Pareto optimal solutions are also weakly Pareto optimal, but not vice versa. This may be observed in the points x_1 and x_2, which are both weakly Pareto optimum, but only x_1 is also Pareto optimal $f_2(x_1) < f_2(x_2)$, but $f_1(x_1)$=$f_1(x_2)$ Pareto front or Pareto optimum set refers to the set of all strictly Pareto optimal solutions in the objective space. To solve a multi-objective optimization issue, the ranges of all objective functions in the objective space should be known. Individually minimizing the function f_i yields the lower bound $z^* \in Z$ for the objective function $f_i(x)$. When all of these lower limits are added together to form a

single vector,$z^* = [z_1^*, \cdots, z_k^*]$the vector is called ideal objective vector. The nadir objective vector is a set of upper boundaries for objective functions in the objective space[6].

Improving one of the objectives will have a negative impact on at least one other. A Pareto dominating vector, also known as a non-inferior or non-dominated vector, is the corresponding objective vector $\vec{f}(\vec{x})$. The Pareto optimal set is the collection of all Pareto optimal solutions. An analytical formulation of the Pareto front is quite tough to come up with.

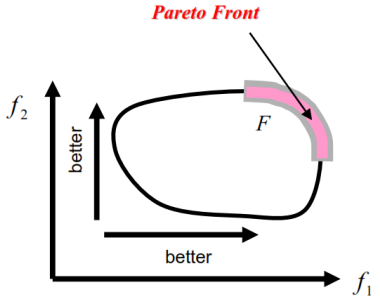


Figure 4: maximizef_1, f_2[10]

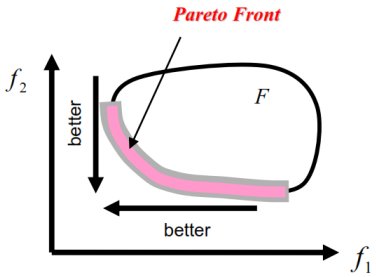


Figure 5: minimizef_1, f_2[10]

2.1 Multi-objective concept in Controller Tuning

Multiple conflicting objective functions are minimized or maximized under restrictions in multi-objective optimization. Analysis of design trade-offs, selection of an ideal product or process design, and any other application requiring an optimal solution with trade-offs between two or more competing objectives are examples of challenges.

It's important to note that the task of designing a PEMFC stack's temperature management involves multiple different objectives. As a result, multi-objective optimization is particularly well suited to this task.

The majority of engineering design statements, especially controller tuning, may be expressed as an optimization issue. First we should define decision variables $\theta = [\theta_1, \cdots, \theta_n]$ and design objectives $J = [J_1(\theta), \cdots, J_m(\theta)]$ Consider the following PI controller:

$$u(t) = k_c \left(e(t) + \frac{1}{T_i} \int_0^t e(t)dt \right)$$

The proportional gain K_c and the integral time T_i will be the decision variables for the PI tuning issue. That is $\theta = [K_c, T_i]$. Assume you choose the Integral of the Absolute Error (IAE), which is the cumulative difference between intended and managed output:

$$J_1(\theta) = IAE(\theta) = \int_{t=t_0}^{t_f} |r(t) - y(t)|dt = \int_{t=t_0}^{t_f} |e(t)|dt$$

Also the tuning problem:

$$minJ_1(\theta) = minIAE(\theta)$$

$$st:$$

$$\underline{\theta_i} \leq \theta_i \leq \bar{\theta_i}$$

The lower and higher boundaries of the choice variables are $\underline{\theta_i}$ and $\bar{\theta_i}$, respectively.Clearly, the solution found and its performance are highly dependent on the design goal.

If a different objective is set, for example:

$$minJ_2(\theta) = min \int_0^t |\frac{du(t)}{dt}|dt$$

$$st:$$

$$\underline{\theta_i} \leq \theta_i \leq \bar{\theta_i}$$

The results achieved by each solution in terms of $J_1(\theta)$ and $J_2(\theta)$ are optimal for the purpose for which they were computed, but not when compared to the other.This situation might make some question:

- Which solution is better?
- Which controller tuning should be implemented for the given process?

Both solutions are based on the problem's practical elements. There isn't a better option than the other; rather, there are solutions with varying trade-offs between (seemingly) contradictory goals. Finally, the implementation of controller parameters K_c and T_i will be determined by the designer's preferences and the process needs. If one of the solutions meets the designer's requirements, the tuning problem is handled by applying the set of parameters from one of the aforementioned optimization problems. However, the designer may be interested in concurrently minimizing both $J_1(\theta)$ and $J_2(\theta)$ objectives, i.e., a solution with a different exchange between those competing objectives. There is no one ideal solution since several objectives are defined, and the objectives are in competition (no one is better than any other[8].

3 PID Controller Design Concept

The concept of feedback is quite strong. Its application has frequently resulted in dramatic changes in performance. Often, credit is given to a certain type of feedback, despite the fact that it is regular input that provides the real advantages, and the type of feedback provided is mostly immaterial. By far the most popular type of feedback in use today is the PID controller. PID control loops account for more than 90% of all control loops. Because derivative action isn't employed very often, most loops are PI. The past (I), present (P), and future (D) control errors are used to provide integral, proportional, and derivative feedback. It's amazing how much can be accomplished with such a basic method. The PID controller's strength is that it addresses major practical difficulties including actuator saturation and integrator windup. As a result, the PID controller is the backbone of automated control. When using feedback, it is the first solution that should be tried. Process control, motor drives, magnetic and optical memory, automotive, flight control, instrumentation, and other applications employ the PID controller. Standard single-loop controllers, software components in programmable logic controllers and distributed control systems, and built-in controllers in robots and CD players are all examples of the controller.

Although the PID controller has always been important, theoreticians have only shown a sporadic interest in it. As a result, many critical concerns have gone undocumented in the literature. As a result, numerous mistakes were repeated as technology progressed from pneumatic to electrical to digital. However, in the last 10 years, there has been a surge of interest. The rise of automated tuning is one cause; another is the rising use of predictive model control, which necessitates well-tuned PID controllers at the main level. Most single-loop management publications, however, employ PID controllers with Ziegler–Nichols tuning as a benchmark. This is a highly undesirable condition since the Ziegler–Nichols criteria have a reputation for producing poorer results in many circumstances.[1]

3.1 Proportional (P) Control

Proportional control is one of the actions employed in PID controllers. Feedback control is a type of proportional control. In a closed-looped system, it is the most basic kind of continuous control. Although P-only control reduces process variable volatility, it does not always get the system to the intended set point. It responds faster than most other controllers, allowing the P-only controller to reply a few seconds faster at first. The reaction time

discrepancy might compound as the system gets more complicated (i.e. more complex algorithm), allowing the P controller to reply even a few minutes quicker. Despite the fact that the P-only controller has a faster response time, it causes departure from the fixed point. The offset is the name for this deviation, and it is typically undesirable in a process. The presence of an offset indicates that the system could not be kept stable at the specified set point. It's similar to the experimental errors in a calibration graph, where the line is constantly prevented from crossing the origin by a fixed, constant inaccuracy. Combining P-only control with another type of control, such as I- or D-control, helps reduce the offset. It is crucial to note, however, that the offset, which is implicitly incorporated in each calculation, cannot be totally eliminated.[12]

$$u(t) = K_c e(t)$$

where k_c is the proportional gain and the feedback error is the difference between the reference signal $r(t)$ and the output signal $y(t)(e(t) = r(t) - y(t))$. The block diagram for the closed-loop feedback control configuration is shown in Figure 1.1 where $R(s)$, $E(s)$, $U(s)$, and $Y(s)$ are the Laplace transforms of the reference signal, feedback error, control signal, and output signal, respectively. $G(s)$ represents the Laplace transfer function of the plant.

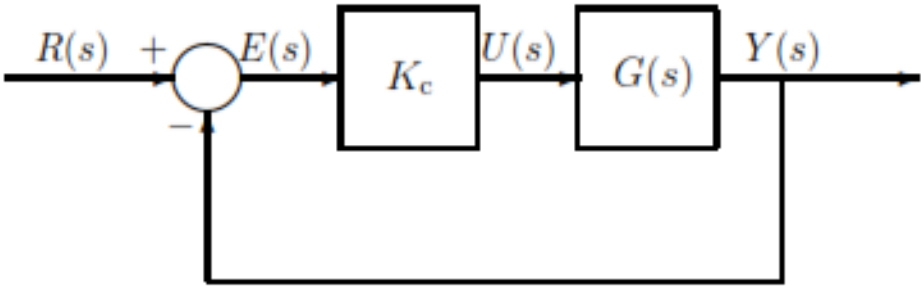


Figure 6: Proportional feedback control system.[13]

3.2 Integral (I) Control

To get a very exact and smooth reaction in plant output, merely considering the immediate error is not an amazing alternative. Because the proportional control has no records of previous experiences, such as the evolution of the reaction over time, it shows the aforementioned shortcomings. To eliminate the error at steady-state, a control action proportional to the integral of the error is applied. The integral of the error is correlated to the controller output through I-control. With regard to time, the integral of the error is computed. It's the sum of all errors during a certain period of time. Equation below demonstrates this I-control behavior numerically.

$$c(t) = \frac{1}{T_i} \int e(t)dt + c(t_0)$$

where

- $c(t)$ is the controller output
- $e(t)$ is the error
- T_i is the integral time
- $c(t_0)$ is the controller output before integration

3.3 Derivative (D) Control

Finally, the third component of the PID controller comes into the scene to reduce response oscillations. The two other components have demonstrated their ability to produce a fast and precise response but at the expense of causing oscillations and high overshoots. This is because in both cases, the controller has no information about the change of the error, but only its instant value and its integral along the time. The derivative of the error provides valuable information about how the error value is going and acts to establish it on a constant value. The derivative control will act firmly when there are strong changes in the error, typically during the slopes of the error.[2]D-control links the controller output to the error's derivative. The error's derivative is calculated with regard to time. It's the variation in error as a function of time. Equation below depicts this D-control behavior numerically.

$$c(t) = T_d \frac{de}{dt}$$

3.4 Proportional-Integral (PI) Control

The PI-control, which lacks the D-control of the PID system, is one example of a combination. A type of feedback control is PI control. Due to the addition of proportional action, it has a faster response time than the I-only control. PI control can prevent the system from oscillating and return it to its fixed point. Even while the PI-control has a better response time than the I-only control, it is still up to 50% slower than the P-only control. As a result, PI control is frequently used in conjunction with D-only control to improve reaction time. The controller's output is linked to the error and the integral using PI-control. This PI-control behaviour is depicted mathematically

$$c(t) = K_c(e(t) + \frac{1}{T_i}\int e(t)dt) + C$$

- $c(t)$ is the controller output
- $e(t)$ is the error
- T_i is the integral time
- K_c is the controller output before integration
- C is the initial value of controller

The integral time is the time it takes the I-only section of the controller to match the control supplied by the P-only part in this equation. According to the equation, the PI-controller works as a simplified PID-controller with a zero derivative term. The PI-controller can alternatively be thought of as a hybrid of the P-only and I-only control equations. The integral action of the I-only control is equivalent to the bias term in the P-only control. When the system is not at the specified point, the P-only control is the sole option. The error equals zero when the system reaches the set point, and the first term is removed from the equation.The system is then managed solely by the controller's I-only section. P-only control will be used if the system deviates from the established point again.[12]

3.5 PID Controller

A combination of these three types of control approaches is known as proportional-integral-derivative control. Because it incorporates the benefits of each form of control, PID-control is the most often employed. The P-only control results in a faster reaction time, while the combined derivative and integral controllers result in a decreased/zero offset. Using the I-control in addition to the offset was used to eliminate it. When utilized in conjunction, D-control considerably improves the controller's reaction since it predicts system disruptions by detecting the change in error. When opposed to the faster P-only control, it has a slower reaction time when used alone, as previously indicated.

$$c(t) = K_c(e(t) + \frac{1}{T_i}\int e(t)dt + Td\frac{de}{dt}) + C$$

3.6 The Windup Phenomenon

In an industrial facility, nonlinearity can manifest itself in a variety of ways. Due to the possibility that the process plant is nonlinear, changing operating circumstances will result in distinct process models and dynamics. The most typical technique for maintaining acceptable control performance throughout a wide variety of nonlinear operating circumstances is to schedule a series of PID controllers, each of which is intended to achieve high performance at a single operating point. Gain or controller scheduling options are available on many PID controller devices. The nonlinear behavior of process actuators presents a unique challenge for PID control. The input and output activities of many of these actuator devices are restricted. Valves, for instance, have a completely open position, a fully closed position, and a flow characteristic that might be linear or nonlinear.

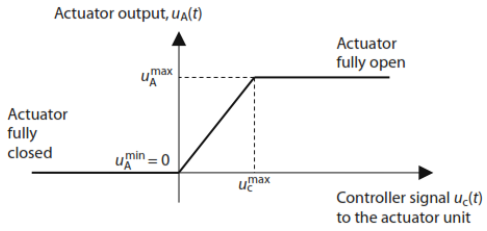


Figure 7: Typical actuator saturation characteristic.[4]

Plot (a) in Figure 11 depicts the input switch step starting at +1 at $t = 0$ and moving to –1 at $t = 5$. The I-control signal $u_c(t)$ is shown in plot (b), and the actual actuator output signal $u_A(t)$ is shown in plot(c). The control signal $u_c(t)$ ramps up to a peak of 10 at $t = 5$ when the step input is at +1, as shown in plot(b). This is when the crucial term comes to a close. In the same time interval $0 \leq t \leq 5$, the actuator output $u_A(t)$ in plot (c) has reached saturation, and the process is being driven by a continuous step signal. When the step input is set to –1 at time $t = 5$, the I-control signal $u_A(t)$ starts to decline in value, and the integral term begins to unwind, as illustrated in plot (b). Despite this, the actuator output $u_A(t)$ stays in the saturation area for the most of the unwind period, and the process is still controlled by a continuous step input signal (plot (c))[4].

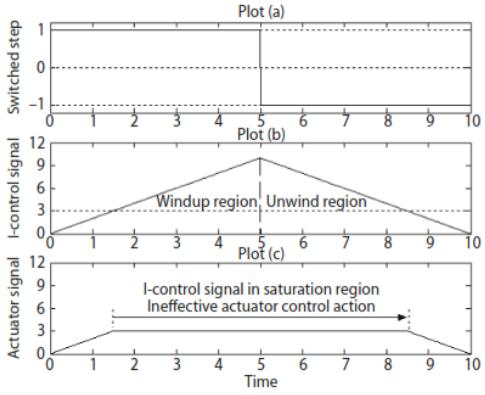


Figure 8: Input switched step, I-control and actuator signals..[4]

3.6.1 Anti windup

Integrator windup can be avoided by turning off the integral action when the control signal reaches the saturation zone and turning it back on when the controller re-enters the linear control region. An anti-windup circuit is used to achieve this switching. An anti-windup circuit is incorporated in most commercial PID controllers, although the circuit specifics are generally not disclosed to the end-user. Simply said, several engineers have applied their brains to this challenge and devised various anti-windup circuit designs. Some of these designs are exclusive to specific controllers on the market.

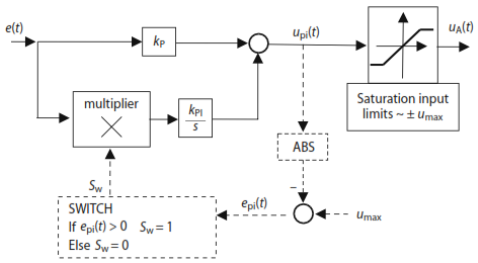


Figure 9: anti-windup circuit for PI control.[4]

The principle of the circuit is quite simple:

- The saturation input limits are denoted u_max
- Compute $e_{pi}(t)$=$u_max - |u_{pi}(t)|$
- If $e_{pi}(t) > 0$ then $S_w = 1$ else $S_w = 0$.

The value of S_w is used to turn on and off the integral term in the PI controller. Simple simulations indicate that this anti-windup circuit effectively eliminates excessive overshoot produced by the control action being in the saturation area for an extended length of time.

3.7 Feed-forward control concepts

One of the most essential parts of control issues is disturbances. In reality, there would be no need for feedback if there were no disturbances, process uncertainty, or unstable dynamics. Load disturbances enter the control loop at some time throughout the operation and cause the system to deviate from its intended operating point. Low-frequency load disturbances are common, and efficient load disturbance reduction is a key issue in process control systems.Disturbances have traditionally been dealt with in a roundabout way, such as by inserting appropriate action into the feedback controller. Feed-forward control from measurable disturbances allows you to execute control actions before the process output responds to the disturbance. As a result, it's an excellent complement to the feedback controller. In the early applications, feed-forward was almost a prerequisite to solving the control problems in these complex applications. Nowadays, feed-forward is implemented in most distributed control systems, and the technique is also used. The feed-forward compensator's design is based on a fairly simple concept. With the sign inverted, the ideal compensate is generated by dividing the

dynamics between the load disturbance and the process output by the dynamics between the control signal and the process output. The impacts of the load disturbance are removed from the process output when this feed-forward compensator is utilized. The perfect compensate, on the other hand, is rarely attainable. The compensator might be non-causal, unstable, have an infinite high-frequency gain due to a derivative action, and need a more sophisticated structure than is necessary. The feed-forward compensator structure in in-process control is often either a gain or a lead-lag filter. A wait is sometimes necessary to guarantee that the compensation is not paid too soon. Compensate structures that are more sophisticated are unusual. They frequently note the challenges with reliability as well, but they seldom go on to give design guidelines. Consider the Laplace transformations of

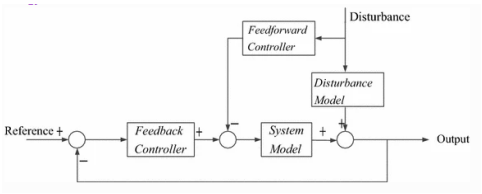


Figure 10: Block diagram of feed-forward-feedback control system[5]

the reference set-point, process output, and disturbance as Y_{sp}, $Y(s)$, $D(s)$, the feedback and feed-forward controllers as $G_c(s)$ and $G_f(s)$, and the process model and disturbance model transfer functions as $G_p(s)$ and $G_d(s)$, respectively, as shown in Fig. 10. The following is the closed-loop transfer function from disturbance to process output:

$$\frac{Y(s)}{D(s)} = \frac{G_d(s) - G_f(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

Ideally, we prefer to achieve perfect control, so the effect of disturbance should be eliminated entirely by setting it to zero. Hence, the feed-forward controller $G_f(s)$ is got as:

$$G_f(s) = \frac{G_d(s)}{G_p(s)}$$

As we mainly focus on industry processes in this paper, we just take some common used controlled plants into consideration. consider:

$$G_d(s) = \frac{K_d}{\tau_d s + 1}, G_p(s) = \frac{K_p}{\tau_p s + 1}$$

where, K_p, K_d are steady states gains, τ_p, τ_d are time constants of these two models respectively. The ideal feed-forward controller[5]:

$$G_f(s) = (\frac{K_d}{K_p})(\frac{\tau_p s + 1}{\tau_d s + 1})$$

.

3.8 Decoupler Design

Multi-input, multi-output (MIMO) systems are used in the majority of industrial operations. Because of the interconnections between the input/output variables, controller design for MIMO processes is more complicated than for single-input, single-output (SISO) processes. Because the control loops interact, one loop's tuning cannot be done alone. Decentralized (multiloop) controllers, decoupled controllers, and centralized controllers can all be used to control the MIMO process. In multiloop control, MIMO processes are viewed as a collection of multi-single loops. The controller is built and implemented on each loop while taking into account the interactions. Multi loop controllers are frequently utilized because of their superior performance, simplicity, and resilience. When the interactions between the loops are small, decentralized controllers perform well. Centralized controllers are preferable if the interactions are significant. It is tough to create a controller for each loop separately.Consider the two-input two-output (TITO) systems with the decentralized control system as shown in Fig. 11. $G(s)$ and $G_c(s)$ are process transfer function matrix and decentralized controller matrix of TITO systems respectively and are represented as $G(s) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$ and $G_c(s) = \begin{bmatrix} g_{c11} & 0 \\ 0 & g_{c22} \end{bmatrix}$

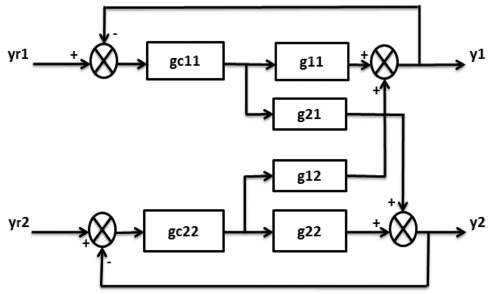


Figure 11: Simplified decentralized control system of a TITO process.[11]

The process transfer function models are expressed:

$$g_{ij}(s) = \frac{k_{pij}e^{-(-\theta_{ij}s)}}{(\tau_{ij}s+1)} i = 1, 2$$

j=1,2 This input-output relation can be expressed:

$$Y(s) = G(s)U(s)$$

$$Y(s) = \begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} and U(s) = \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

where Y(s) and U(s) are the output and input vectors. The TITO system's input output connection may be represented :

$$y_1(s) = g_{11}(s)u_1 + g_{12}(s)u_2(s)$$

$$y_2(s) = g_{21}(s)u_1 + g_{22}(s)u_2(s)$$

In the TITO system, when the second loop is closed, the input from u_i to y_i has two transmission paths. The combination of two transmission paths is considered as effective open-loop dynamics. If the second feedback controller is in the automatic mode, with $y_{r2} = 0$, then the overall closed-loop transfer function between y_1 and u_1 is given by

$$\frac{y_1}{u_1} = g_{11} - \frac{g_{12}g_{21}g_{c2}}{1 + g_{c2}g_{22}}$$

This can be written as

$$\frac{y_1}{u_1} = g_{11} - \frac{g_{12}g_{21}(g_{c2}g_{22})}{g_{22}(1 + g_{c2}g_{22})}$$

and also

$$\frac{y_2}{u_2} = g_{22} - \frac{g_{12}g_{21}(g_{c1}g_{11})}{g_{11}(1 + g_{c1}g_{11})}$$

we can also consider:

$$\frac{g_{ci}g_{ii}}{1 + g_{ci}g_{ii}} = 1 \, and \, i = 1, 2$$

$$g_{11}^{eff} = \frac{y_1}{u_1} = g_1 1 - \frac{g_{12}g_{21}}{g_{22}} \, while \, g_{22}^{eff} = \frac{y_2}{u_1} = g_2 2 - \frac{g_{12}g_{21}}{g_{11}}$$

Here $g_1 1^{eff}$ and $g_2 2^{eff}$ are the effective open-loop transfer function[11].

3.9 Operating Point

A dynamic system's operational point describes the model's starting states and root-level input signals at a certain time.Operating points can be discovered using command-line tools, the Steady State Manager, or by linearizing a model with the Model Linearizer. State variables that do not vary with time are included in a model's steady-state operating point, also known as an equilibrium or trim condition. There can be several steady-state functioning points in a model. A hanging damped pendulum, for example, has two steady-state working points where the position of the pendulum does not fluctuate over time. When a pendulum hangs straight down, it reaches a stable steady-state working point. The pendulum always returns to balance when its position deviates little. Small changes in the operating point, in other words, do not cause the system to depart from the area of excellent approximation around the equilibrium value.When a pendulum points upward, it creates an unstable steady-state operating point. The pendulum is in balance as long as it points directly upward. The operating point swings downward when the pendulum deviates slightly from this location, leaving the region around the equilibrium value. To achieve convergence while using optimization search to compute nonlinear system operating points, your starting predictions for states and input levels must be close to the intended operating point. Choosing the proper operating point when linearizing a model with several steady-state operating points is critical. A stable linear model is produced by linearizing a pendulum around the stable steady-state operating point. In contrast, an unstable linear model is created by linearizing around the unstable steady-state operating point.

3.10 The effects of Degradation on Control system

There are two basic paths in which equipment degradation analysis has historically progressed. Consideration that degradation causes fundamental changes to a system's behavior has been one strategy that has generated a lot of interest for control system design. Another method, used to production scheduling and optimum maintenance planning, views degradation as a distinct process that influences performance but does not always alter behavior. A new, uniform categorization is suggested to achieve this. It considers both the variables generating degradation as well as how degradation affects the behavior of the system. Control system optimization, design, and operation will all be enhanced with an understanding of these interdependent factors. In order to model degradation inside a control system, it is

necessary to describe how degradation affects the system's capacity to carry out its function and how degradation is affected by the way the system is run. Physical variables like temperature and humidity, as well as modes of operation like the sequence of recipes in a batch process, are influencing variables.The operation of the system has a significant impact on degradation, with some operational points likely to hasten decline. In order to limit degradation and lower the likelihood of an unplanned halt prior to a scheduled maintenance overhaul, it may thus be advantageous to adjust the active set points. Therefore, a study of the effects of degradation would be beneficial for the system.[14]

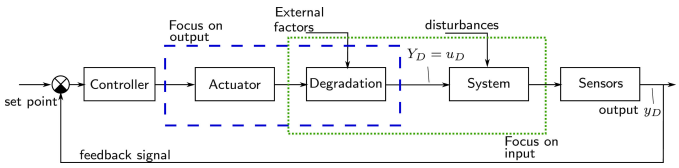


Figure 12: Degradation-dependent control system showing input and output degradation[14]

In a multiplicative model of degradation, the degraded value of a system variable is scaled according to a degradation function $h(d)$ as V_D=$h(d)V$. In particular, $h(d) = (1 - d)$ yields:

$$V_D = (1 - d)V$$

which can be rearranged to a form called the relative model of degradation

$$d = \frac{V - V_D}{V}$$

where $V - D$ denotes the degraded value, V is the value without degradation, d is from the model of degradation and $h(d)$ is a function of the degradation that reflects the effect of degradation on V.

4 TEMPERATURE CONTROL DESIGN

4.1 Methodology

The methodology followed in article consists of three stages:

1)Obtaining a PID based control structure tuned with linear model and validate objective values in nonlinear model

2)Adding feed-forward control to linear model and analyse it and validate objective values in nonlinear model

3)Designing a decoupler for linear model and analyse it and validate objective values in nonlinear model

With using the System Identification Toolbox of matlab,obtained the transfer functions of the linearized model,Which is a 2 × 3 MIMO model, valid around the specified operating point[9]:

$$\begin{bmatrix} T_{w_{out}} \\ T_{w_{in}} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \end{bmatrix} \begin{bmatrix} u_{T_{w_{out}}} \\ u_{T_{w_{in}}} \\ i \end{bmatrix}$$

$$G_{11}(s) = \frac{-0.942}{1 + 24.0s} e^{(-41.6s)}$$

$$G_{12}(s) = \frac{-0.741(1 + 354.0s)}{(1 + 438.7s)(1 + 72.7s)(1 + 13.2s)} e^{(-7.7s)}$$

$$G_{13}(s) = \frac{0.092(1 + 258.9s)}{(1 + 336.1s)(1 + 47.3s)}$$

$$G_{21}(s) = \frac{0.196(1 + 104.6s)}{1 + (2 * 0.934 * 34.1s) + (34.1s)^2} e^{(-41.1s)}$$

$$G_{22}(s) = \frac{-0.772(1 + 358.9s)}{(1 + 409.8s)(1 + 68.9s)(1 + 3.1s)} e^{(-3.8s)}$$

$$G_{23}(s) = \frac{0.050}{1 + 133.6s}$$

The control structure of C_L and C_{NL}(the control designed which the subscripts L and NL refer to the Linear and nonlinear model) is the same: two PI type controllers with anti windup, One for the control of $T_{w_{out}}$ and the other for the control of $T_{w_{in}}$. The derivative actions were switched off to avoid amplifying the measurement noise and thus protecting the actuators. Therefore, each control ($C_L$and C_{NL}) has four parameters to adjust (two for each PI), namely K_{c1} $((l/min)/C)$, $K_{c2}((l/min)/C)$,$T_{i1}(s)$ and $T_{i2}(s)$.

These parameters are adjusted by solving a multi-objective optimization problem.

$$u(t) = K_c[e(t) + \frac{1}{T_i}\int e(t)dt]$$

It's critical to understand that no single solution in a Pareto set is better than another in the same set for all of the objectives. It's important to note that each of these optimal solutions is a controller, i.e. a specific modification of the parameters of the two PI controllers that make up the control structure. As a result, we'll refer to solutions as well as controllers from now on. The a posteriori multi-objective optimization process has the following advantage: at the final decision-making step, the designer is aware of all optimum solutions and can thus directly compare them. This enables them to examine the trade-off between the many Pareto set solutions and, as a result, pick one of them knowing all necessary facts, increasing the designer's confidence that the final chosen solution is the correct one.

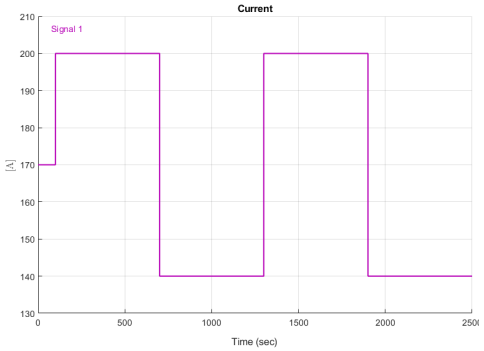


Figure 13: Electric current demand signal used in the design of the controls. The first step is applied at t=100 s and the span of each step is 600 s, so changes occur in 100, 700, 1300 and 1900 seconds

Both of the PI controllers' settings are adjusted at the same time. This is crucial because there is a tight connection between the control actions and the outputs because of the nature of the process: $u_{T_{w_{out}}}$ and $u_{T_{w_{in}}}$ affects both $T_{w_{in}}$ and $T_{w_{out}}$. These coupling effects are accounted for implicitly in the control's design by concurrently altering both PI controllers. Two crucial design goals are keeping $T_{w_{out}}$ and $T_{w_{in}}$ points($65°C$ and $60°C$, respectively) at their prescribed temperatures. These two goals are intended to produce maximum electrical efficiency and minimal stack degradation. A crucial component that must be considered in the control design is con-

trol efforts, in addition to reducing the output errors relative to their set points. There are two causes for this. First, the actuators' power consumption decreases the system's overall electrical efficiency, which is dependent on control efforts. Second, overly aggressive control operations can considerably shorten the lifetime of the actuators (valves, pumps). Consequently, control efforts will be included in the MOP with two new objectives. Note that the difficulty of developing the temperature regulation of a PEMFC stack comprises numerous competing objectives. For this reason, the multi-objective optimization methodology is especially appropriate for this task. It is feasible to describe the control design as a multi-objective optimization problem once constraints and objectives have been determined, as in the following example:

$$minf(x)$$

where:

$$f(x) = [f_1(x)\ f_2(x)\ f_3(x)\ f_4(x)]$$

and

$$x = [K_{c1}\ K_{c2}\ T_{i1}\ T_{i2}]$$

subject to:

$$\underline{x} \leq x \leq \bar{x}$$

$$\underline{x} = [-5\ -5\ 1\ 1]$$

$$\bar{x} = [-0.1\ -0.1\ 100\ 100]$$

Objectives:

$$f_1(x) = \frac{1}{T_{sim}} \int_0^{T_{sim}} |e_{T_{w_{out}}}(t)|\, dt$$

$$f_2(x) = \frac{1}{T_{sim}} \int_0^{T_{sim}} |e_{T_{w_{in}}}(t)|\, dt$$

$$f_3(x) = \frac{1}{T_{sim}} \int_0^{T_{sim}} |\frac{du_{T_{w_{out}}}(t)}{dt}|\, dt$$

$$f_4(x) = \frac{1}{T_{sim}} \int_0^{T_{sim}} |\frac{du_{T_{w_{in}}}(t)}{dt}|\, dt$$

The objective f_1 is the average absolute error in the stack temperature $T_{w_{out}}$, in $^{\circ}C$. The objective f_2 is the average absolute error in stack inlet water temperature $T_{w_{in}}$, $^{\circ}C$. The objective f_3 is the average absolute value of the rate of change of the control action $u_{T_{w_{out}}}$, in$(l/min)/s$. The objective f_4 is the average absolute value of the rate of change of the control action

$u_{T_{w_{in}}}$, in $(l/min)/s$. T_{sim} is the simulation time ($2500s$). The optimization algorithm used for the search of the optimal solutions is evMOGA[7]. evMOGA Multiobjective Evolutionary Algorithm has been developed by the Predictive Control and Heuristic optimization Group (CPOH) at Universitat Politecnica de Valencia (Spain). The values for the parameterization of the algorithm are: eMOGA.NindP=100 (Individuals for the P population), eMOGA.NindGA=8 (individuals for aux population GA and should be an even number), eMOGA.Generations=60 (number of generations).

It is essential to put the correct set-point to maintain temperatures. All transfer function in the model work with real variables ($T_{w_{out}} = 65°C$,$T_{w_{in}} = 60°C$, $I = 170A$, $u_{T_{w_{out}}} = 4.43l/min$ and $u_{T_{w_{in}}} = 4.99l/min$). If we want to obtain the system around the operating point, the set-point for model should be $0°C$, and it perfectly would work because nothing moves. fig 13. And also the electric current demand signal works around -30 and 30.

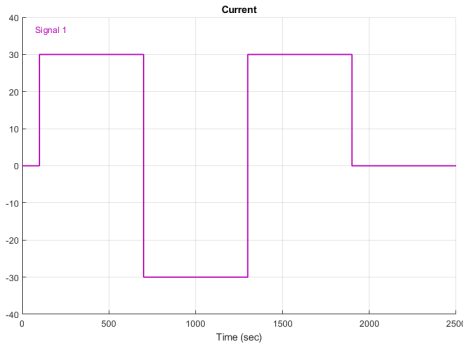


Figure 14

Anti-windup is a way to avoid a problem with the integrator and limitation of variables if the system has saturation. Therefore we need to change PID with an offset input (PID-CPOH) that we usually use for feed-forwards and decoupling. Without anti-windup, all harmful errors make integral reduce not to control the system.

4.2 Results

1)obtaining and analysing controllers with linear model and validate objective values in nonlinear model

The results of optimal controller using the linear model are shown in Table 1.

Table 1: Optimal controller obtained using the linear model

	K_{c1}	K_{c2}	T_{i1}	T_{i2}
X1	-0.54853	-4.05155	54.89263	34.09732
X2	-0.50204	-3.95333	51.25941	44.81721
X3	-0.47819	-3.97427	52.82471	42.49065
X4	-0.4633	-3.97923	52.3481	48.58409
X5	-0.4303	-3.96544	50.69075	48.86343
X6	-0.41728	-3.94847	50.53998	52.17404
X7	-0.42238	-3.92455	51.63982	55.29795
X8	-0.37847	-3.93907	48.19238	53.96708
X9	-0.35563	-3.85732	45.89953	56.77104
X10	-0.34898	-3.92231	47.4241	57.9841
X11	-0.32998	-3.91313	46.12663	58.75471
X12	-0.2911	-3.90882	44.38295	59.59017
X13	-0.26832	-3.91405	43.06856	60.02207
X14	-0.23172	-3.93974	40.69315	61.28303
X15	-0.22012	-4.06291	53.17865	42.82286
X16	-0.2331	-4.19621	70.10664	16.70356
X17	-0.21579	-4.21039	70.43927	18.29688
X18	-0.20337	-4.22346	71.49656	16.68616
X19	-0.19215	-4.2217	71.27896	16.75243

We aggregate objectives to simplify the problems that J_1=$f_1(x)$+$f_2(x)$ and J_2=f_3+f_4 and Q_1 and Q_2 have the same formula with J_1 and J_2 but for nonlinear model, respectively. They are the validations of the parameters value of solution X_i in the nonlinear model in table 2. It also shows the values of the four objectives for each of the X_i solutions of C_L, in two scenarios: 1) when these solutions are simulated using the linear model (columns J_1 and J_2) and 2) when they are simulated using the nonlinear model (columns Q_1 and Q_2) and the average percentage change (controller

performance degrades when simulated using the nonlinear model) ($J\%$).

$$J_1 = \frac{|J_{1lin} - Q_{1nlin}|}{|J_{1lin}|} * 100\%$$

$$J_2 = \frac{|J_{2lin} - Q_{2nlin}|}{|J_{2lin}|} * 100\%$$

$$J\% = \sqrt{J_1^2 + J_2^2}$$

Waveform distortion and random noise both lead to system performance deterioration, although they may be examined independently to make system design and performance evaluation simpler.

Table 2: Values of the objective functions for the solutions of C_L and validate it in nonlinear model

	J_1	J_2	Q_1	Q_2	J%
X1	0.344557	0.002445	0.423041	3.85E-03	42.96129
X2	0.353333	0.002118	0.448776	3.68E-03	48.02524
X3	0.360815	0.00201	0.470193	3.62E-03	49.8844
X4	0.371314	0.001886	0.48492	3.36E-03	46.19408
X5	0.381942	0.001782	0.5062	3.47E-03	52.68956
X6	0.393482	0.00171	0.520758	3.42E-03	53.41476
X7	0.401003	0.001689	0.531147	3.59E-03	58.9354
X8	0.411588	0.001605	0.540892	3.22E-03	50.82349
X9	0.424296	0.001556	0.556822	3.23E-03	52.62172
X10	0.438408	0.001513	0.578208	3.42E-03	59.49663
X11	0.449468	0.00148	0.590459	3.13E-03	52.3192
X12	0.480339	0.001426	0.628823	3.04E-03	51.69151
X13	0.500076	0.0014	0.65392	3.22E-03	57.64282
X14	0.537723	0.001368	0.696991	3.02E-03	53.26402
X15	0.629821	0.001345	0.805647	3.17E-03	58.82389
X16	0.650317	0.00131	0.846871	4.17E-03	88.71998
X17	0.680595	0.001296	0.877361	3.75E-03	77.25136
X18	0.700634	0.001256	0.906283	4.01E-03	86.06826
X19	0.717803	0.001245	0.932849	4.51E-03	100.9246

In figure 15, the x-axis shows performance, and the y-axis shows control action and the objective value of the linear and nonlinear models with

circle and square, respectively. As it turns out, the linear model has better performance and control action one. Every solution has a performance degradation that is on average more than 40%. Therefore, when the solutions found using the linear model are checked against the nonlinear model, it can be said (in this situation) that the controllers' performance typically degrades (which represents the real plant).

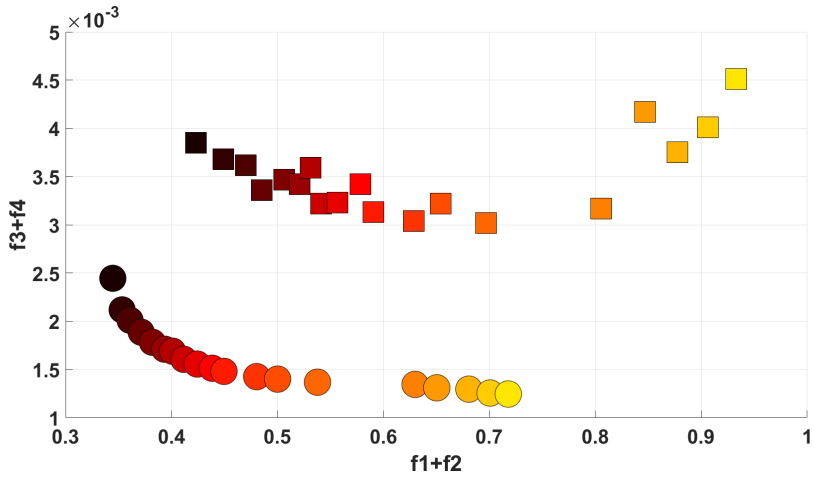
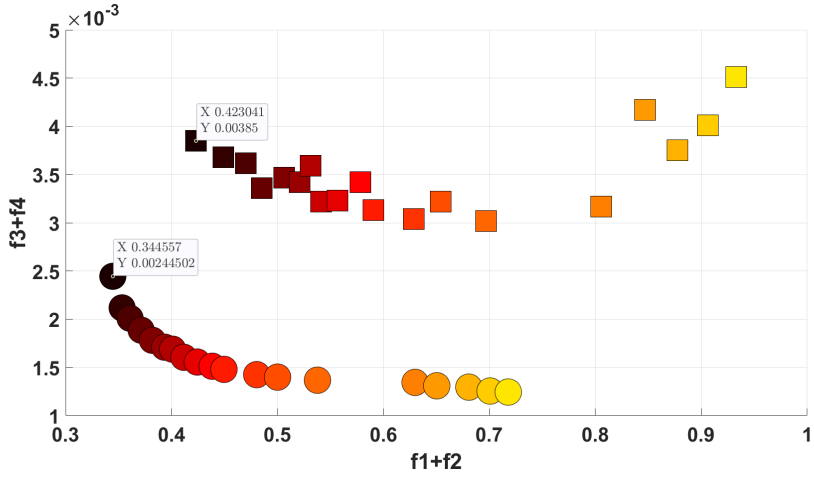
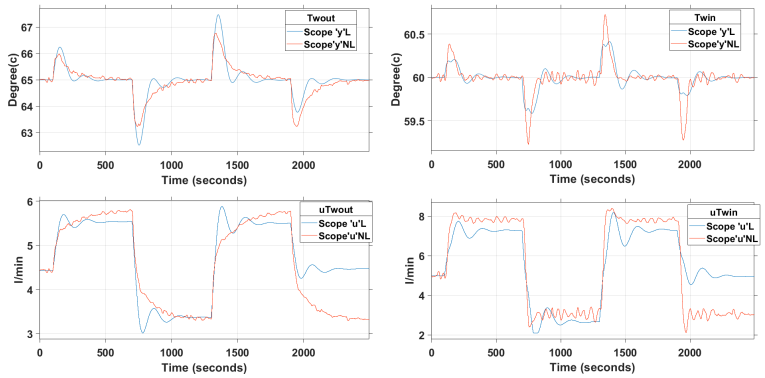


Figure 15: Values of the objective functions for the solutions of C_L and validate it in nonlinear model

In figure 16 (a). we can see one of the extremes, the objective value of the $X1$ solution. It has the fewest degradation compared to other solutions. In figure 16(b), It is visible that it has the best performance for system output and the worst control action compared to other solutions. Therefore we can imagine that when the graph is more aggressive, it has better performance and worse control action and vice versa. And also we can observe that In the second output ($T_{w_{in}}$), the overshoot of the nonlinear model is greater than the linear one, while the opposite has happened in the first output. In control action plots, it is clear that for $u_{T_{w_{out}}}$, linear and nonlinear graphs have a huge difference together. In the linear model, we have overshot, and in 500s, it goes to stable, while in the nonlinear model, the model will gradually be stable but fluctuate. Moreover, the simulated response is satisfactory in the $u_{T_{w_{out}}}$ plot. But both plots have a substantial difference in the two models after 1900s.



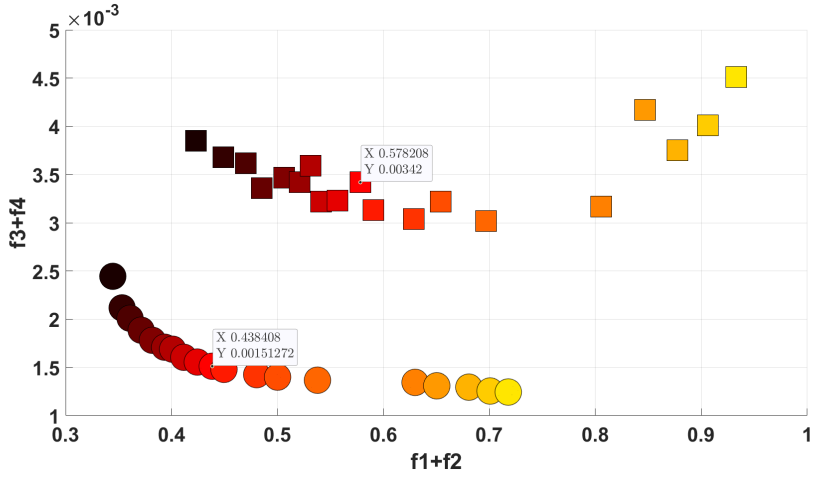
(a)



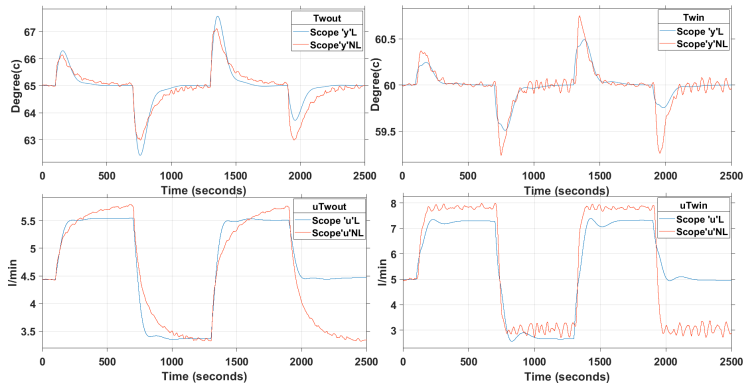
(b)

Figure 16: Experimental validation of solutions $X1$

In figure 17, a solution from the middle of the Pareto front ($X10$) has been chosen. It is shown that the overshoot in both outputs of the nonlinear and linear model is increased. The degradation has been changed to 59.4%. Two linear output graphs are smoother, but the performance is worse. And it turns out that linear and nonlinear control action will be more compatible. In general, the settling time is going to decline.



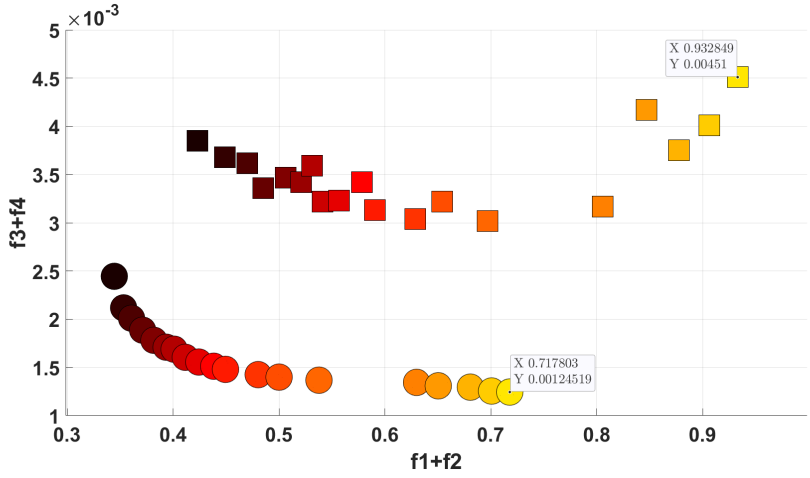
(a)



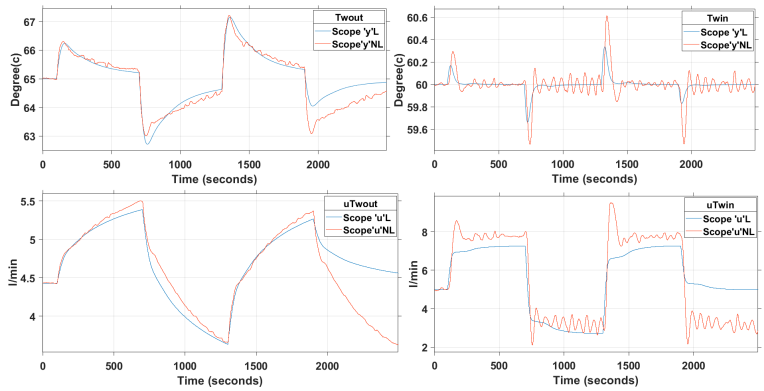
(b)

Figure 17: Experimental validation of solutions $X10$

the most degradation of controller is related to the $X19$ with 100.9%(figure 18). The performance degradation suffered by the solution $X19$ is such that the real response of the system controlled by making more time to allow outputs to reach refference. This means that solution $X19$ (optimal solution when designing the control using the linear model and, as a result, a solution which could have been chosen by the designer) is completely unacceptable, because of its features (a lot of errors, settling time,$\cdots$). Both cases, there is a very good match between the simulated response and the experimental response. But the noise in nonlinear model of both $T_{w_{in}}$ and $u_{T_{w_{in}}}$ Has intensified.



(a)



(b)

Figure 18: Experimental validation of solutions $X19$

2)Adding feed-forward to PID based control tuned with linear model and validate objective values in nonlinear model

The idea is to put a gain as feed-forward, showing $K1$ and $K2$ for two PI. Feed-forward is the best control strategy when the cause of regular disturbances is recognized or well defined. Feed Forward Control may plan an effective reaction by simulating the range of disturbances originating from the source after Feed Forward has located it. And also, there is a limitation for these two parameters between 0.01 to 0.2 that we reached to be sure that many solutions are out of saturation limit.

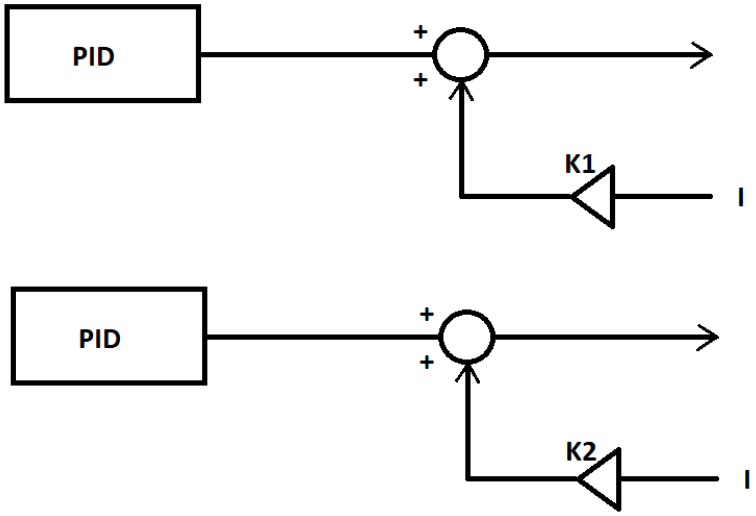


Figure 19: Adding feedforward to PID controller

Table 3: Parameter optimization results obtained using the linear model with feed-forward

	K_{c1}	K_{c2}	T_{i1}	T_{i2}	$K1$	$K2$
X1	-0.41799	-3.28993	63.36079	56.43983	0.025097	0.091721
X2	-0.42677	-2.20495	55.07361	60.48287	0.022354	0.087042
X3	-0.41191	-2.25013	55.10677	60.41569	0.021621	0.08678
X4	-0.39781	-2.31204	55.00343	60.37643	0.020596	0.08626
X5	-0.38634	-2.26783	55.10638	60.47249	0.019706	0.084699
X6	-0.38172	-2.27103	54.99439	60.32323	0.019294	0.084751
X7	-0.37459	-2.24411	54.98476	60.56536	0.018247	0.083282
X8	-0.36663	-2.28589	55.54929	60.71694	0.017848	0.082867
X9	-0.33923	-2.28791	55.90884	60.81586	0.017494	0.082841
X10	-0.31841	-2.3096	56.45801	60.68178	0.017322	0.082341
X11	-0.29074	-2.29988	56.59697	60.86304	0.016891	0.082327
X12	-0.26948	-2.31899	56.92437	60.86023	0.016508	0.081874
X13	-0.23876	-2.32042	57.55821	61.01217	0.016121	0.081654
X14	-0.20511	-2.36484	58.11211	61.04079	0.015443	0.081011
X15	-0.16731	-2.38086	58.83327	61.3802	0.014718	0.08088
X16	-0.17395	-2.41393	59.41154	61.28434	0.01406	0.079805
X17	-0.11537	-2.43541	59.80787	61.28231	0.014118	0.08002
X18	-0.1	-2.42995	59.83821	61.21145	0.01333	0.080066
X19	-0.1	-2.36371	59.57691	61.58364	0.012383	0.079157

In order to reduce the issues caused by F_1=$f_1(x) + f_2(x)$, F_2=$f_3 + f_4$, and D_1 and D_2 having the identical formula as F_1 and F_2, respectively, we aggregate our goals. The parameters value of the solution Xi in the nonlinear model in table 4 has been validated.

Table 4: Values of the objective functions for the solutions of C_L and validate it in nonlinear model with feed-forward

	F1	F2	D1	D2	F%
X1	0.181626	0.004331	0.25814	0.005788	51.80%
X2	0.18873	0.003735	0.261235	0.004778	37.96%
X3	0.193874	0.003674	0.276391	0.004859	43.16%
X4	0.204118	0.003598	0.28595	0.004654	38.94%
X5	0.214078	0.003476	0.299409	0.004585	40.87%
X6	0.218836	0.003451	0.306035	0.004628	43.18%
X7	0.229512	0.003344	0.318025	0.004462	41.34%
X8	0.237004	0.003314	0.329428	0.004478	43%
X9	0.255564	0.003265	0.354017	0.004486	45.26%
X10	0.271268	0.003231	0.371674	0.004348	42.09%
X11	0.297534	0.003199	0.401726	0.004273	41.05%
X12	0.321213	0.00317	0.428096	0.004272	42.13%
X13	0.358937	0.003139	0.470092	0.00429	43.95%
X14	0.408091	0.003097	0.524448	0.00405	38.41%
X15	0.470689	0.003055	0.595509	0.003987	38.57%
X16	0.474555	0.003021	0.598629	0.004041	41.04%
X17	0.551216	0.002978	0.694061	0.003871	39.33%
X18	0.586954	0.002939	0.740832	0.003895	42.18%
X19	0.607925	0.00288	0.763344	0.00376	40.37%

The table 4 shows that the performance degradation of the design with feed-forward is less than without it. Therefore it is expected to have better performance which we can see the Pareto front of the models (figure 21) are closer to Origin of coordinates compare to without feed-forwards. By adding the feed-forward the degradation decreased significantly(between 37.96% to 51.8%).

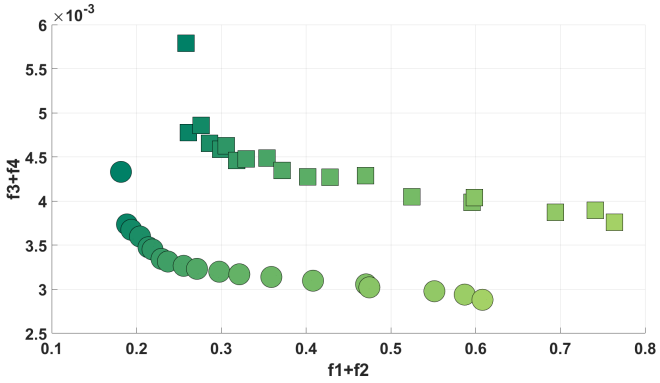


Figure 20: Parameter optimization results obtained using the linear model with feed-forward

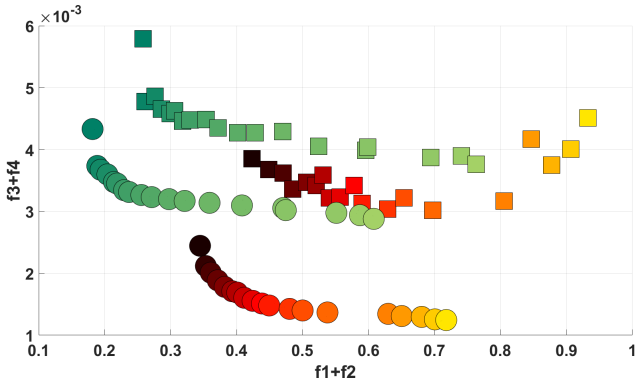
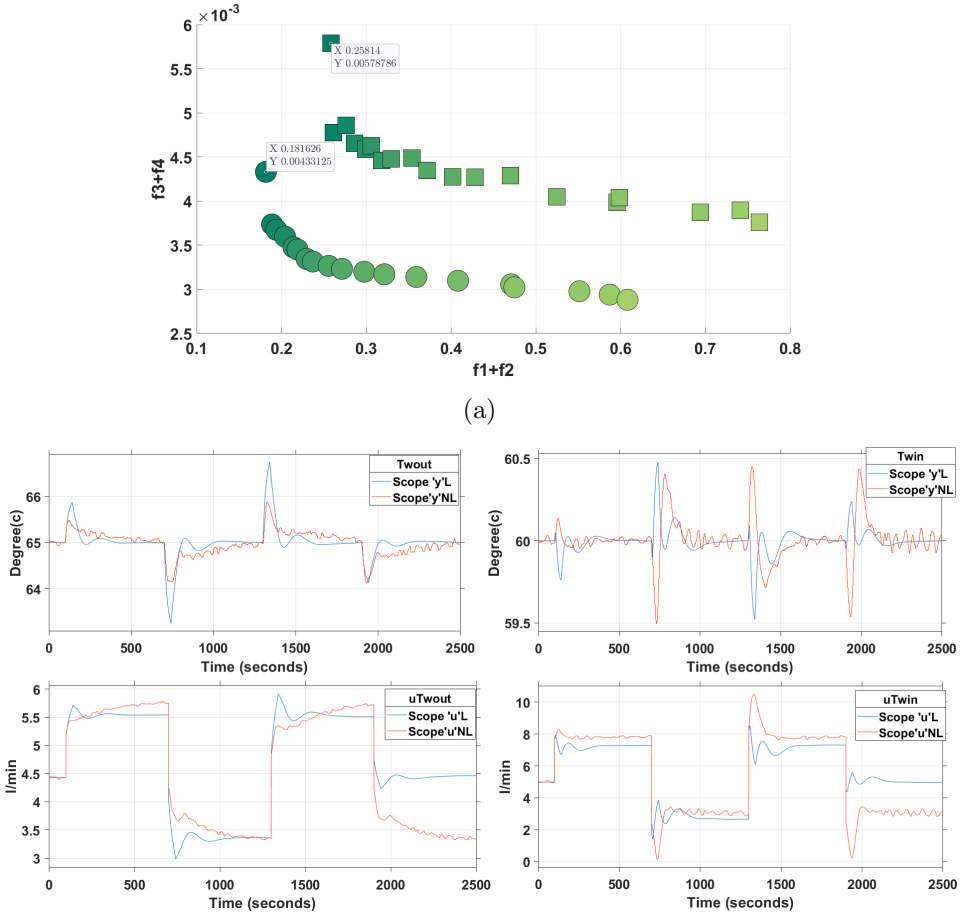


Figure 21: optimal controller obtained using the linear model with and without feed-forward

In figure 22, We check out one of the objective value extremes of models with feed-forward ($X1$). In these models, we can see two outputs of the PI controller will be stable at less than 500s. The overshoot of linear and nonlinear will be less than $1°C$. And also, we can see less noise in control action for the nonlinear model compared to the same model without feed-forward as well as the linear model has been changed with a shorter settling time. The degradation of this solution is the most (51.80%)

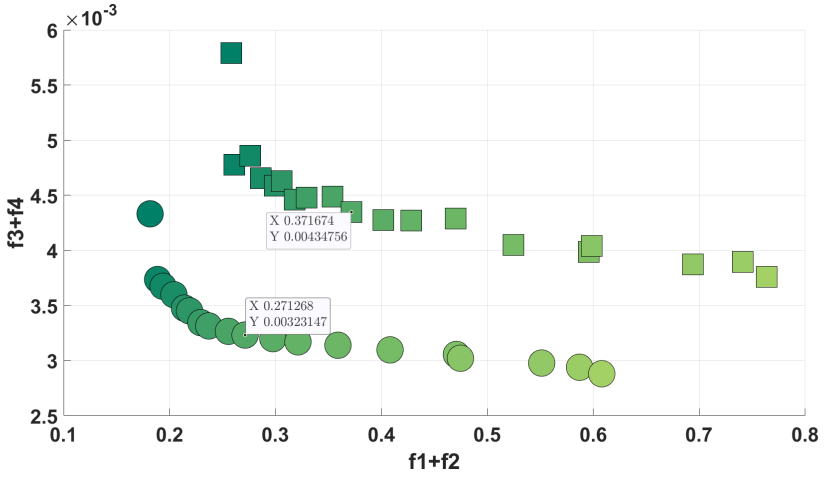


(a)

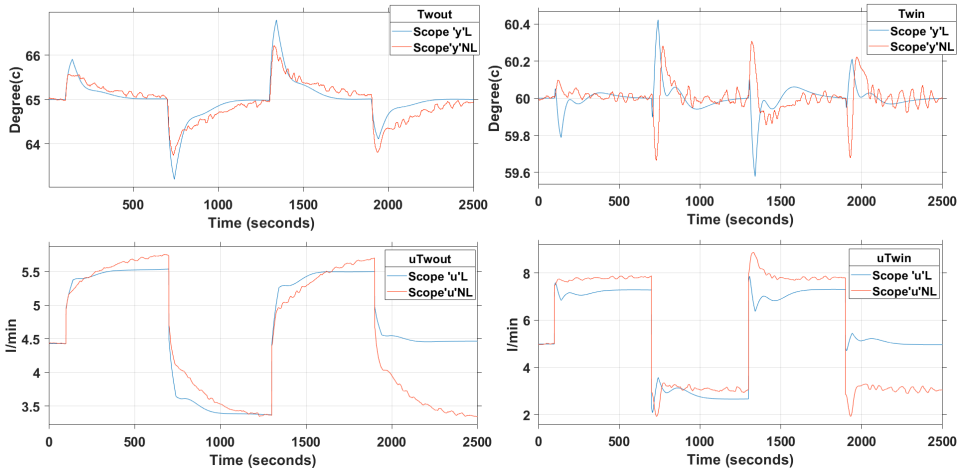
(b)

Figure 22: Experimental validation of solutions with feed-forward $X1$

in figure below the measure degradation has been declined. Both linear and nonlinear model outputs became smoother, while the control action output did not change much for nonlinear, but no error was seen for the linear model. Also, the noise in all the graphs has decreased slightly. Therefore the performance is going to be worse but control action do better.

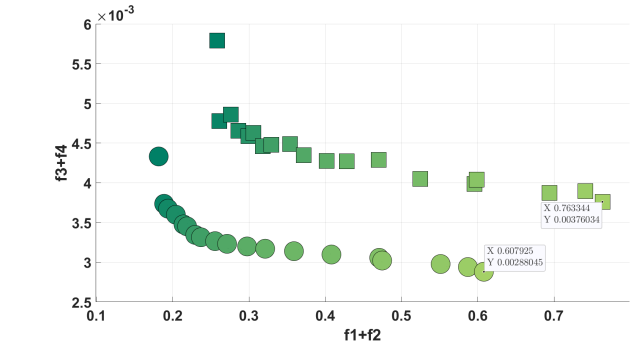


(a)

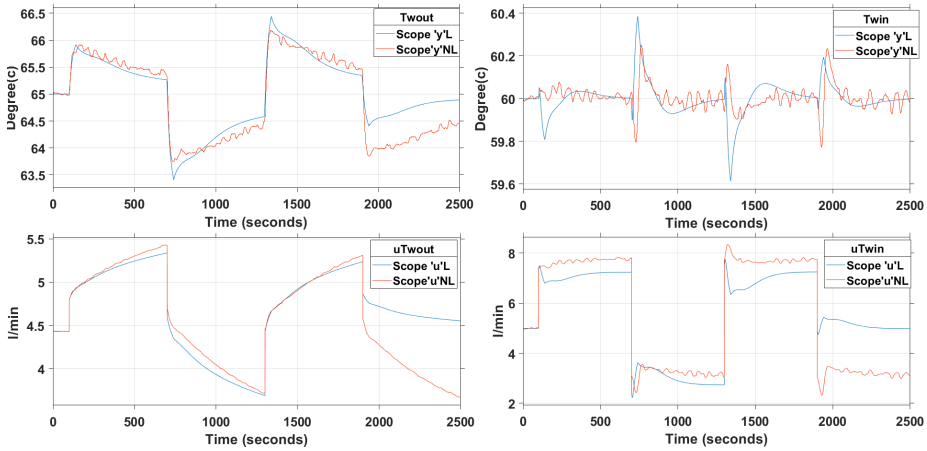


(b)

Figure 23: Experimental validation of solutions with feed-forward $X10$



(a)



(b)

Figure 24: Experimental validation of solutions with feed-forward $X19$

We can see the best compatibility between linear and nonlinear models in system output and control action graphs in figure 24 (solution $X19$). The degradation in these objective values is almost identical to the objective value of solution $X10$ (40.37% and 42.09%, respectively). In this case, we have the least aggressive and the worst performance.

3)obtaining and analysing linear model and validate objective values in nonlinear model with feed-forward and decoupler

In this section we have two additional parameters then it is more possibility of tuning. We are trying to consider the influence of two subsystems with a static decoupler.

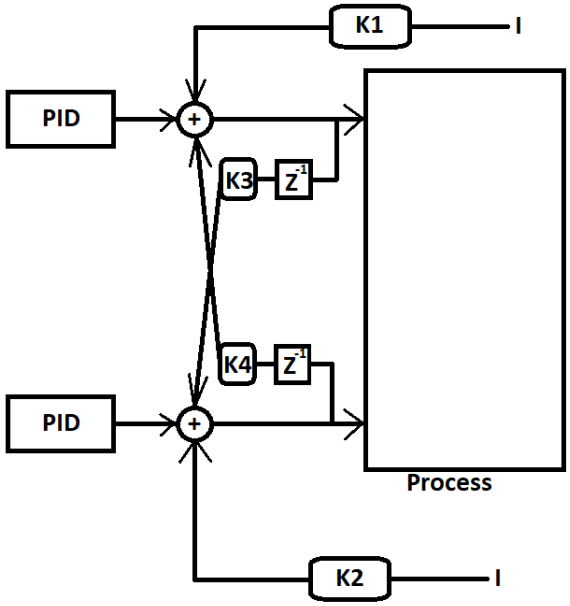


Figure 25: Adding feedforward and decoupler to PID controller

Table 5: Parameter optimization results obtained using the linear model with feed-forward and decoupler

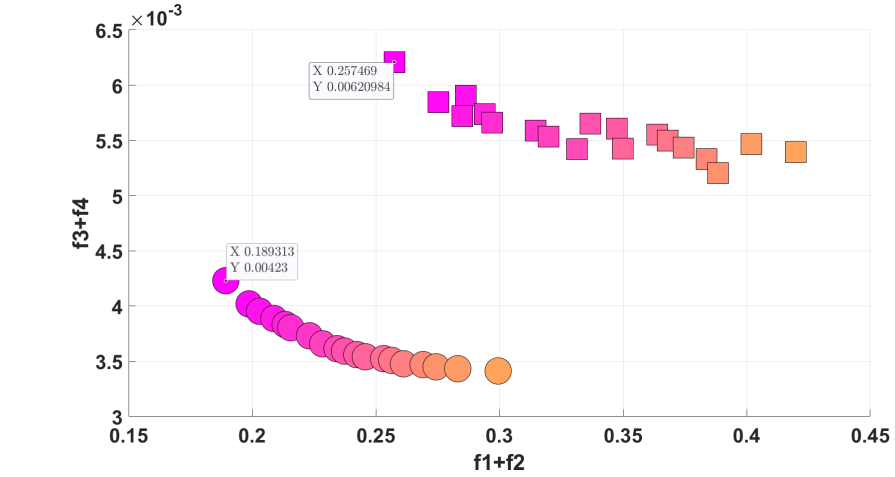
	K_{c1}	K_{c2}	T_{i1}	T_{i2}	$K1$	$K2$	$K3$	$K4$
x1	-0.41583	-4.15908	51.01268	44.99698	0.017289	0.084958	0.053434	0.15447
x2	-0.38716	-4.1416	50.95329	45.36867	0.015997	0.083462	0.054503	0.153878
x3	-0.36844	-4.15147	51.04457	45.44276	0.01611	0.082651	0.054363	0.153488
x4	-0.43311	-4.05292	49.63232	47.19091	0.011776	0.083401	0.058838	0.152471
x5	-0.42792	-4.05298	49.54878	47.4535	0.011196	0.083214	0.05872	0.152696
x6	-0.41847	-4.05823	49.64246	47.36309	0.011272	0.082932	0.058626	0.152611
x7	-0.35583	-4.11722	50.59652	46.22759	0.013425	0.0815	0.056625	0.152512
x8	-0.36696	-4.0793	50.15141	46.93606	0.011935	0.081119	0.057817	0.152236
x9	-0.34989	-4.08531	50.33642	46.79571	0.012044	0.080657	0.057624	0.152212
x10	-0.34074	-4.0892	50.42761	46.71836	0.012149	0.08035	0.057493	0.152149
x11	-0.32597	-4.09766	50.56808	46.60529	0.012354	0.079802	0.057208	0.151999
x12	-0.316	-4.1011	50.65789	46.50534	0.012519	0.079406	0.057085	0.151879
x13	-0.30402	-4.09524	50.75159	46.31195	0.012388	0.079525	0.057324	0.152128
x14	-0.29175	-4.10587	50.86851	46.28084	0.01281	0.078904	0.057061	0.151848
x15	-0.2732	-4.12515	50.9957	46.29526	0.01352	0.077634	0.056802	0.151455
x16	-0.25773	-4.13772	51.1846	46.24299	0.013881	0.07725	0.056408	0.151105
x17	-0.26448	-4.12161	51.18778	46.0736	0.013023	0.07779	0.056396	0.15162
x18	-0.25103	-4.12814	51.30595	45.94224	0.013242	0.077225	0.056155	0.151431
x19	-0.24408	-4.14928	51.28239	46.02754	0.012424	0.077577	0.056464	0.151414

Table 6: Values of the objective functions for the solutions of C_L and validate it in nonlinear model with feed-forward and decoupler

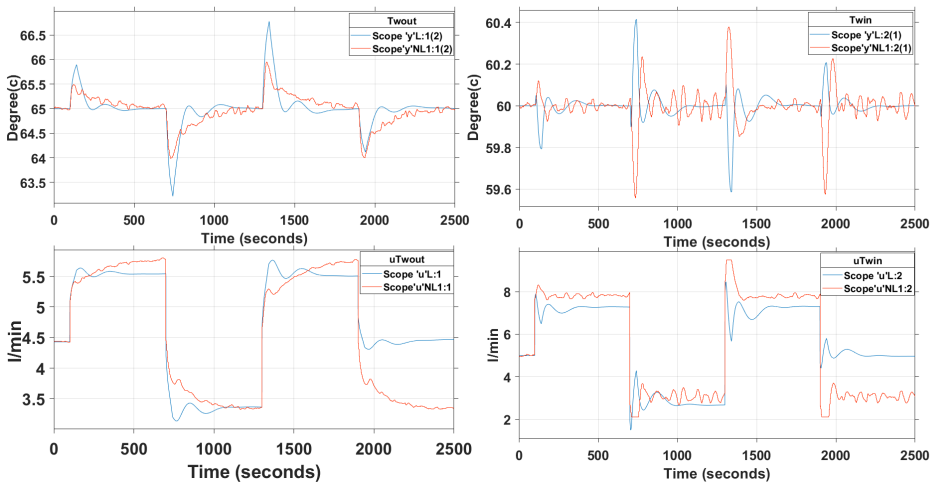
	G1	G2	T1	T2	G%
x1	0.189313	4.23E-03	0.257469	0.00621	70.51737
x2	0.198724	0.00402006	0.275206	0.005848	66.05815
x3	0.202978	0.00395343	0.286431	0.005906	70.68346
x4	0.208753	0.00388818	0.285073	0.005722	66.24134
x5	0.213347	0.00383359	0.293983	0.005738	68.9034
x6	0.215631	0.00380391	0.297173	0.005662	67.4319
x7	0.223145	0.00373124	0.3146	0.005589	68.26731
x8	0.228491	0.00365845	0.319822	0.005534	68.84327
x9	0.23434	0.00361452	0.331357	0.005424	67.20745
x10	0.23735	0.00359307	0.336728	0.005654	75.54745
x11	0.242316	0.00356103	0.347399	0.005606	75.54315
x12	0.245764	0.00354124	0.349834	0.005427	70.32804
x13	0.253124	0.00352429	0.363846	0.005553	75.54899
x14	0.256444	0.00350829	0.368115	0.0055	74.46631
x15	0.261204	0.00347778	0.374448	0.005437	73.58022
x16	0.269138	3.47E-03	0.383785	0.005333	70.7042
x17	0.274411	3.45E-03	0.388433	0.005206	67.27468
x18	0.283159	0.00343318	0.401902	0.005466	76.46867
x19	0.299474	0.00341281	0.419887	0.005393	74.9899

The degradation of whole process in this case is between 66.05% to 76.46%. G_1=$f_1(x)$+$f_2(x)$ and G_2=f_3+f_4 and G_1 and G_2 have the same formula with T_1 and T_2 but for nonlinear model, respectively.

In figure 26 there is one of the extreme (solution $X1$), has the quite same result that we saw before in the same solution for feed-forward.

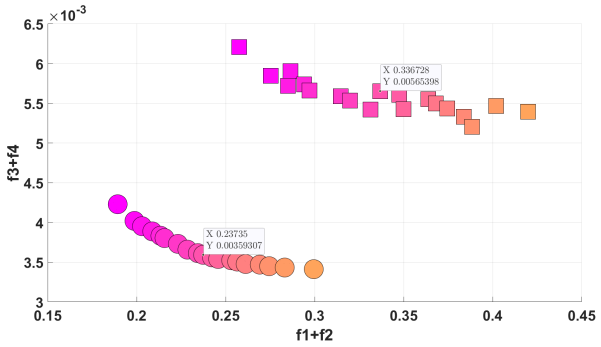


(a)

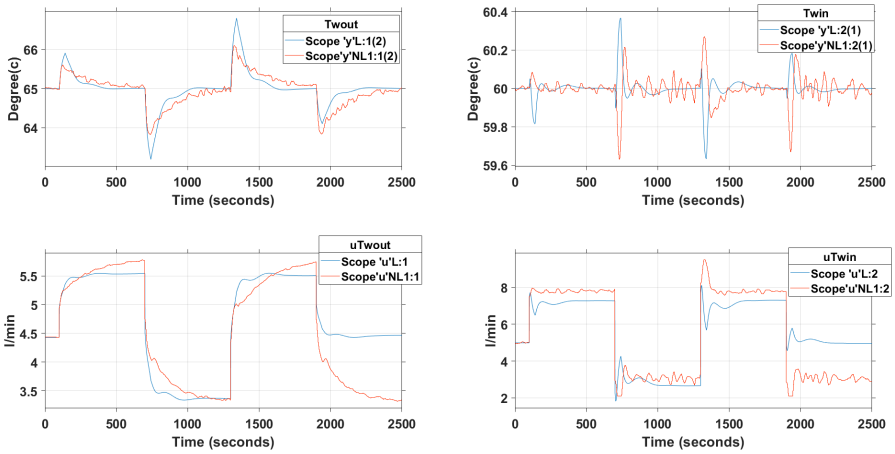


(b)

Figure 26: Experimental validation of solutions with feed-forward and decoupler $X1$



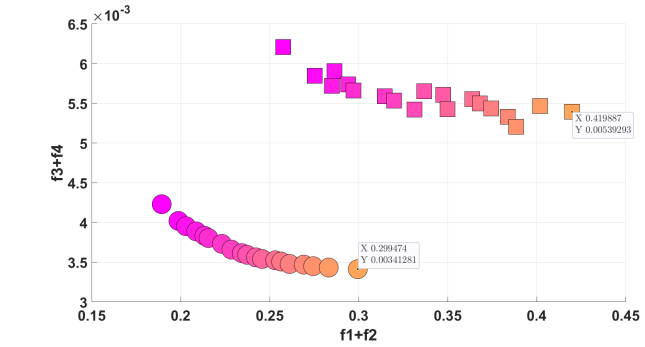
(a)



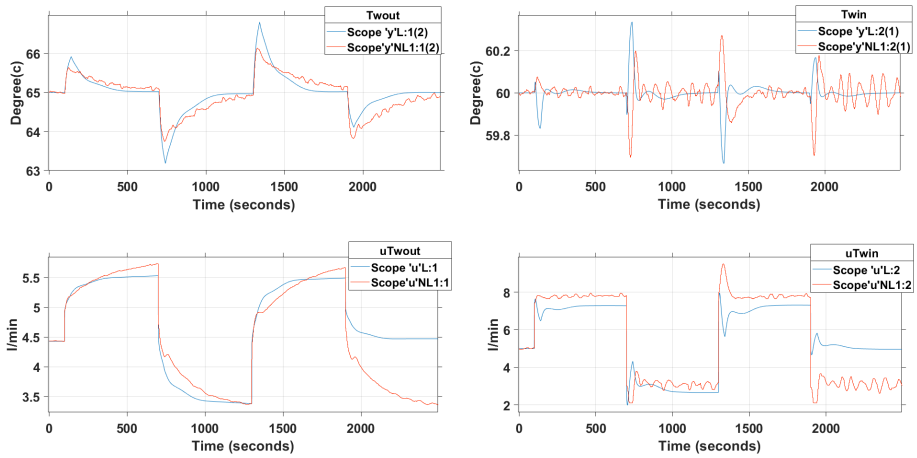
(b)

Figure 27: Experimental validation of solutions with feed-forward and decoupler $X10$

in figure 27, As we expect the outputs (Solution $X10$) are going to be smoother and less control action but the performance of errors are going to be worse.



(a)



(b)

Figure 28: Experimental validation of solutions with feed-forward and decoupler $X19$

smoothest linear and nonlinear model with feed-forward and decoupler belongs to the solution $X19$ which is shown in figure 28. But all results that we reached about decoupler didn't improve the system compare to feed-forward.

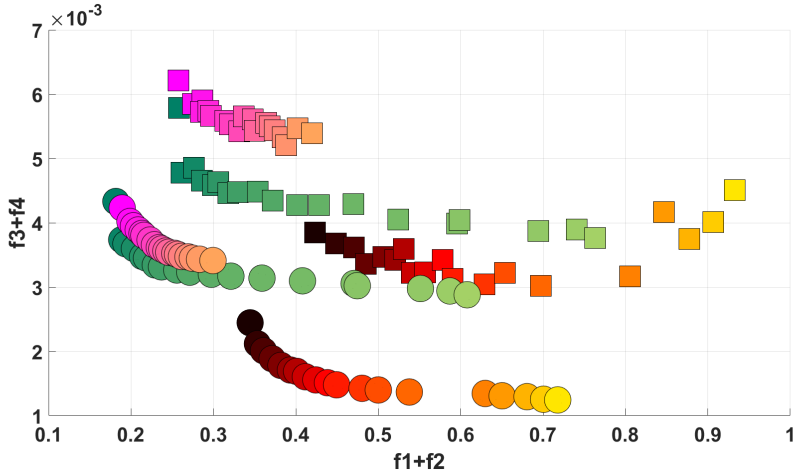


Figure 29: optimal controller obtained using the linear model without and with feedforward and decoupler

We can see the merge of all optimal controller obtained in figure 29,As it can be seen in the figure the all objective value of decoupler design is worse than feedforward design.

5 Conclusion

In control systems, views on performance and control effort differ. A control engineer strives to achieve excellent performance while being aggressive. While the maintenance engineer seeks to increase the lifespan of the equipment. So there should be a good relationship between performance and control effort. As a result, the performance must be sacrificed to some extent in order to improve the quality of control action. The study on multi-objective optimization gives us this possibility. First, the controllers are tuned with linear models(easier to obtain) and then tested/validated on the real plant. In our case, the nonlinear model replaces the real plant. It was observed that the overall performance of the system outputs (input and output temperatures) as well as control actions are consistent, but with high degradation. To compensate for this issue and reduce the impact of disturbance, we used feed-forwards so that we can achieve a better overall performance with less degradation, and the level of error was significantly reduced, but the control of the action was increased. Then, by adding a decoupler to reduce the effect of two controllers on each other, we expected that the overall performance would improve, but unfortunately, not only the quality of the action control did not improve, but the error rate also increased. It can be concluded that maybe adding a process does not always require creating a better overall quality. All the parameters used are easily implemented so that the design cost is significantly low. In some cases, the difference between the linear and non-linear models was quite large. If we have a nonlinear model it is possible to optimize for the nonlinear model instead of using the linear one. But in our real installation probably we don't have a nonlinear model, We have our model and our plant. If we want a nonlinear model we have to pay too much while I was trying to reproduce an industrial behaviour that means I don't have a nonlinear model and by using some common tools tried to obtain a good tune of controller.

References

[1] Karl Johan Åström and Tore Hägglund. The future of pid control. *Control engineering practice*, 9(11):1163–1175, 2001.

[2] J. Canete, Cipriano Galindo, and Inmaculada Moral. *Introduction to Control Systems*, pages 137–165. 01 2011.

[3] Juan J Durillo and Antonio J Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[4] Michael A Johnson and Mohammad H Moradi. *PID control*. Springer, 2005.

[5] Lu Liu, Siyuan Tian, Dingyu Xue, Tao Zhang, and YangQuan Chen. Industrial feedforward control technology: a review. *Journal of Intelligent Manufacturing*, 30(8):2819–2833, 2019.

[6] Jose Malmberg. Data-driven interactive multiobjective optimization using cluster based surrogate in discrete decision space. 2018.

[7] Miguel Martínez-Iranzo, Juan M Herrero, Javier Sanchis, Xavier Blasco, and Sergio García-Nieto. Applied pareto multi-objective optimization by stochastic solvers. *Engineering applications of artificial intelligence*, 22(3):455–465, 2009.

[8] G Reynoso Meza, Blasco Ferragud, Javier Sanchis Saez, and Herrero Durá. *Controller tuning with evolutionary multiobjective optimization*. Springer, 2017.

[9] SANTIAGO NAVARRO, JUAN M HERRERO, XAVIER BLASCO, and RAÚL SIMARRO. Design and experimental validation of the temperature control of a pemfc stack by applying multiobjective optimization.

[10] Patrick Ngatchou, Anahita Zarei, and A El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 84–91. IEEE, 2005.

[11] C Rajapandiyan and M Chidambaram. Controller design for mimo processes based on simple decoupled equivalent transfer functions and

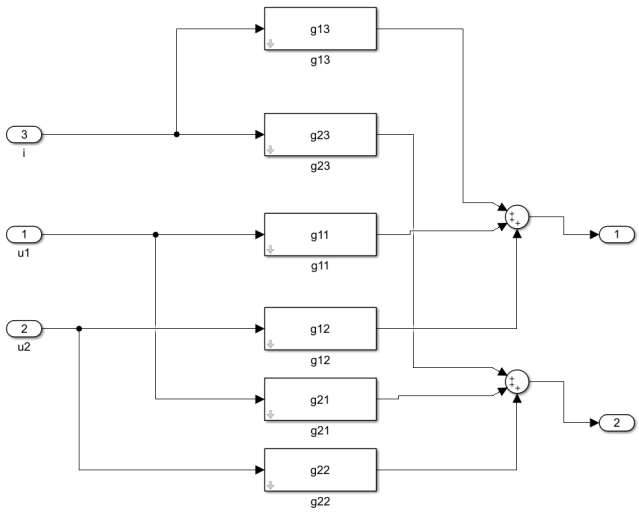
simplified decoupler. *Industrial & Engineering Chemistry Research*, 51(38):12398–12410, 2012.

[12] George Stephanopoulos. *Chemical process control*, volume 2. Prentice hall New Jersey, 1984.

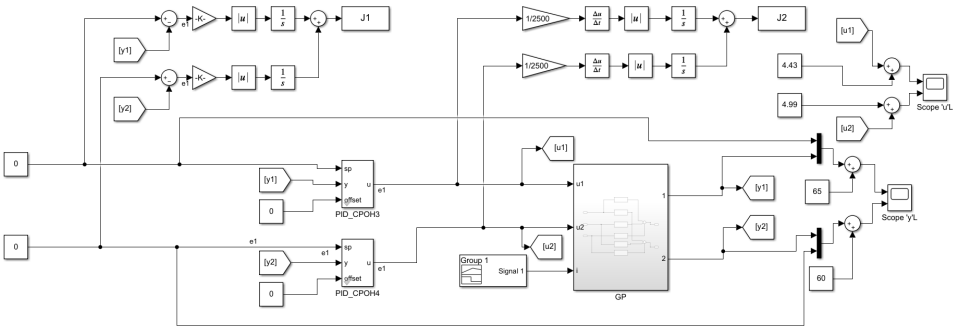
[13] Liuping Wang. *Basics of PID Control*. 03 2020.

[14] Marta Zagorowska, Ouyang Wu, James R Ottewill, Marcus Reble, and Nina F Thornhill. A survey of models of degradation for control applications. *Annual Reviews in Control*, 50:150–173, 2020.

A Matlab simulink



(a)



(b)

Figure 30: Linear model without feed-forward

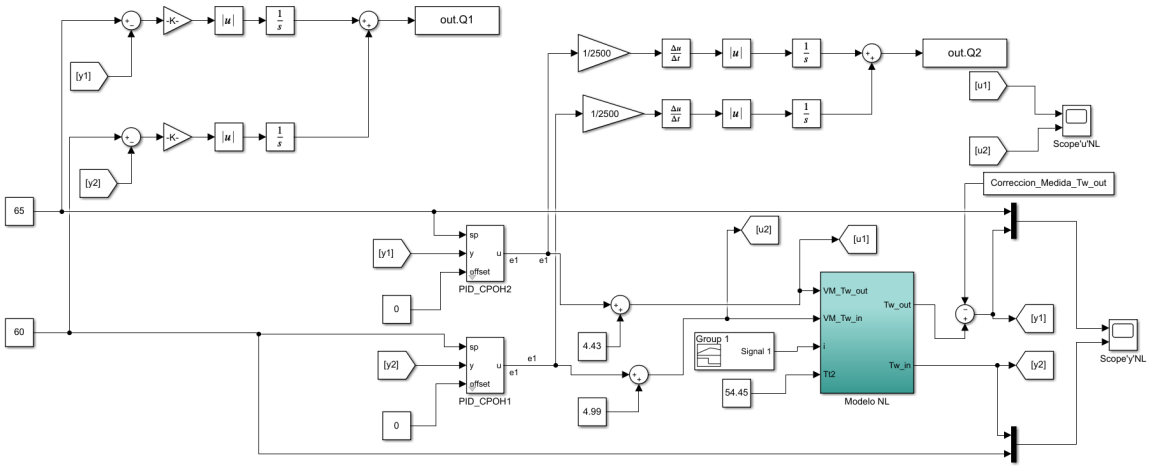


Figure 31: Nonlinear model without feed-forward

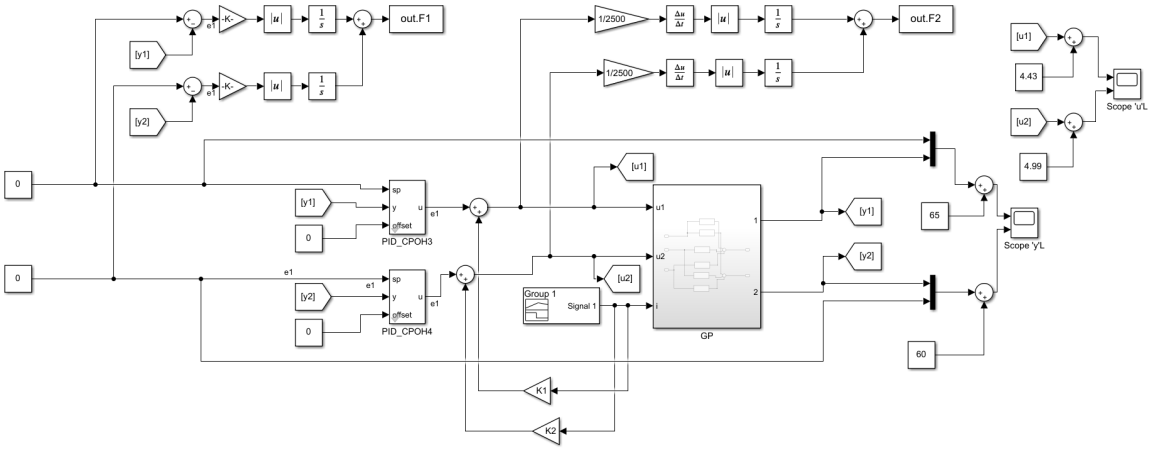


Figure 32: Linear model with feed-forward

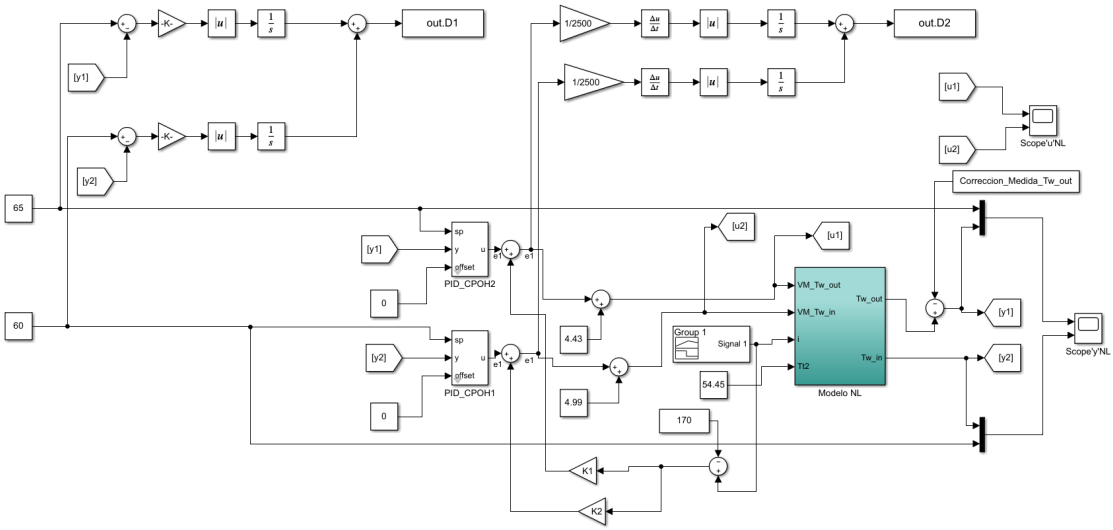


Figure 33: Nonlinear model with feed-forward

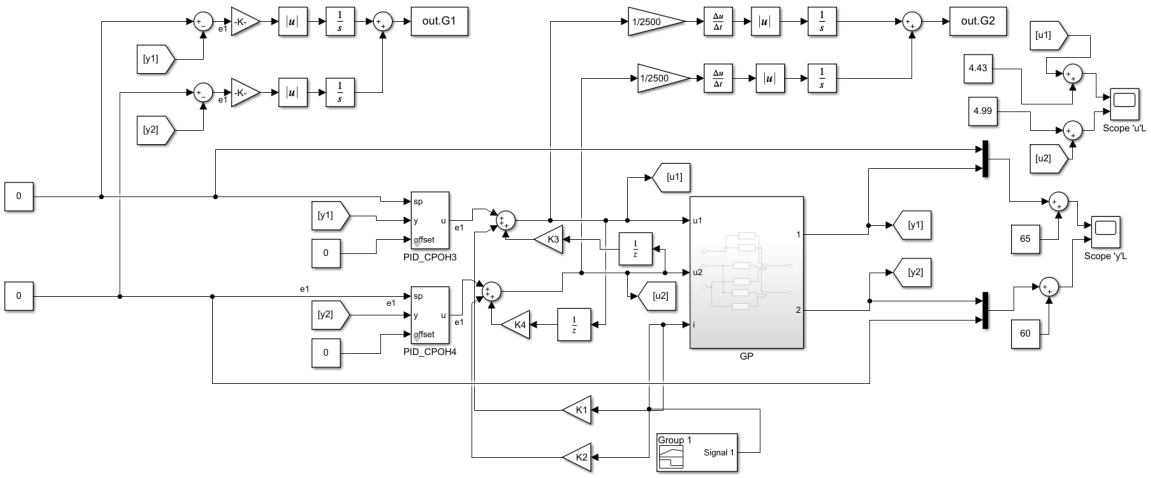


Figure 34: Linear model with feed-forward and decoupler

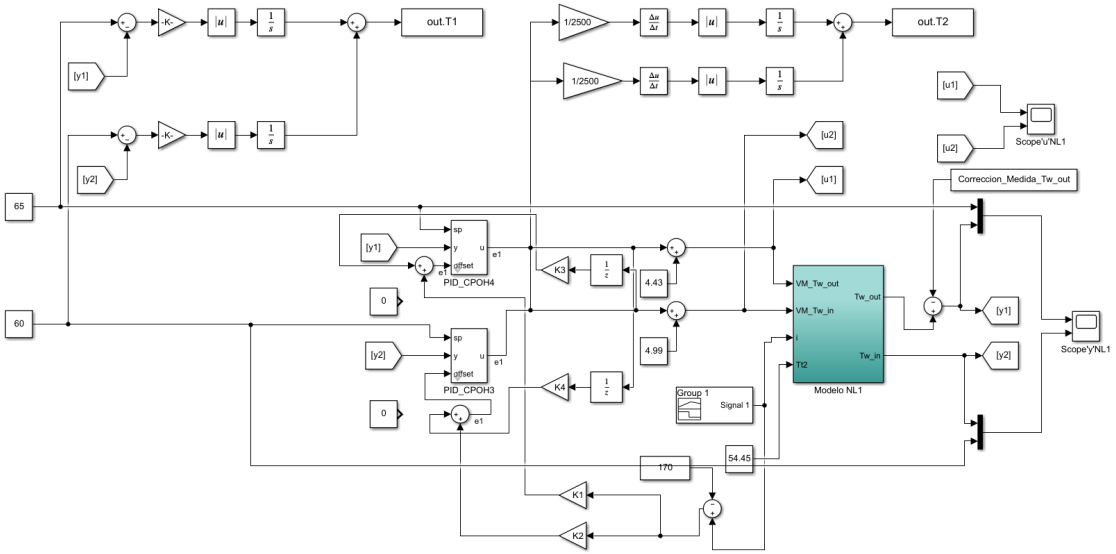


Figure 35: Nonlinear model with feed-forward and decoupler