PID synthesis under probabilistic parametric uncertainty

Mercader, Pedro; Soltesz, Kristian; Banos, Alfonso

Published in:
2016 American Control Conference, ACC 2016

DOI:
10.1109/ACC.2016.7526527

2016

Document Version:
Peer reviewed version (aka post-print)

PID Synthesis under Probabilistic Parametric Uncertainty

Pedro Mercader[1], Kristian Soltesz[2], Alfonso Baños[1]

***Abstract*— In many system identification methods, process model parameters are considered stochastic variables. Several methods do not only yield expectations of these, but in addition their variance, and sometimes higher moments. This paper proposes a method for robust synthesis of the proportional–integral–derivative (PID) compensator, taking parametric process model uncertainty explicitly into account. The proposed method constitutes a stochastic extension to the well-studied minimization of integrated absolute error (IAE) under $\mathcal{H}_\infty$-constraints on relevant transfer functions. The conventional way to find an approximate solution to the extended problem is through Monte Carlo (MC) methods, resulting in high computational cost. In this work, the problem is instead approximated by a deterministic one, through the unscented transform (UT), and its conjugate extension (CUT). The deterministic approximations can be solved efficiently, as demonstrated through several realistic synthesis examples.**

I. Introduction

This paper considers robust synthesis of PID compensators for process models described by transfer functions with a stochastic parametrization $\mathbf{p}$. It is assumed throughout that $\mathbf{p}$ obeys a multivariate Gaussian distribution (although the proposed method can be extended to any distribution with symmetric probability density).

Situations with stochastic parametrization arise naturally when system identification is performed on input–output data. For instance, the use of gradient methods in the output error framework, yield not only estimates of parameter expectations $\mathbb{E}[\mathbf{p}]$, but also estimates of higher order moments, such as the covariance $\mathbb{V}[\mathbf{p}]$ [1]. If the signal-to-noise ratio of the identification data is low (due to high noise levels or short data sets), the parameter covariance significantly impacts the outcome of the synthesis [2].

It is customary to formulate robust PID synthesis as a constrained optimization problem

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & J(s,\mathbf{k},\mathbf{p}), \\ \text{subject to} \quad & \boldsymbol{\varphi}(s,\mathbf{k},\mathbf{p}) \leq \mathbf{0}, \end{aligned} \tag{1}$$

where s is the Laplace variable, $\mathbf{k}$ is the design (compensator) parameter vector, and $\mathbf{p}$ is the stochastic vector of process model parameters. Minimization of the objective J aims at maximizing performance over $\mathbf{k}$, while the constraint vector $\boldsymbol{\varphi}$ (with components φ_i) is in place to ensure robustness. Commonly used objectives and constraints are reviewed in Section II. For now, it is sufficient to note that both J and $\boldsymbol{\varphi}$ are (possibly nonlinear) functions of the stochastic vector $\mathbf{p}$.

Since $\mathbf{p}$ is Gaussian, there is a finite probability of it attaining *any* value, and generally it is not possible to guarantee the constraints of (1). This can be addressed by propagating the uncertainty of the process model parameters $\mathbf{p}$ through to the objective J and constraints $\boldsymbol{\varphi}$, and formulating a stochastic optimization problem

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & \mathbb{E}[J(s,\mathbf{k},\mathbf{p})], \\ \text{subject to} \quad & \mathbb{E}[\varphi_i(s,\mathbf{k},\mathbf{p})] + \alpha\sqrt{\mathbb{V}[\varphi_i(s,\mathbf{k},\mathbf{p})]} \leq \mathbf{0}. \\ & \forall \varphi_i \in \boldsymbol{\varphi} \end{aligned} \tag{2}$$

This formulation is an extension of (1) in the sense that the two formulations are equivalent for deterministic $\mathbf{p}$. The parameter α in (2) enables the user to specify a confidence for constraint fulfillment. While providing a feasible formulation, it is typically not possible to solve (2) exactly, as there exists no explicit way to evaluate the objective or constraints. To see why this is the case, we introduce an arbitrary function g, which takes on the role of J, or one of the components of $\boldsymbol{\varphi}$ in (2).

Given (estimates of) the expectation $\mathbb{E}[\mathbf{p}]$ and covariance $\mathbb{V}[\mathbf{p}]$ (and possibly higher moments) of a stochastic variable $\mathbf{p}$, together with its assumed joint probability density function (PDF) f, we want to evaluate the expectation $\mathbb{E}[g(\mathbf{p})]$, covariance $\mathbb{V}[g(\mathbf{p})]$, and possibly higher moments. It is typically not possible to obtain closed-form evaluations of the corresponding expectation integrals. However, numeric quadrature methods can be used to obtain approximations.

One option is the use of Monte Carlo (MC) quadrature. That is, drawing many samples from the distribution generated by f, applying g to each sample, and computing the ensemble mean, covariance, and possibly higher moments. The drawback of the MC approach lies in its computational burden. In order to produce approximations of adequate confidence, a large number of samples is typically needed.

Another approach, utilized in the extended Kalman filter, and adopted in [3] for robust synthesis, is to approximate g, using a Taylor series expansion. While popular, this method requires computation of the Jacobian (and Hessian) of g.

A third approach, adopted in this paper, relies on the use of test-point methods, further explained in Section III. Rather than approximating g through linearization, these methods evaluate $g(\mathbf{x})$ on a small set of cleverly chosen deterministic test points, which can be pre-computed as they do not depend on g.

[1]Pedro Mercader and Alfonso Baños are with the Department of Computer Sciences and Systems Engineering, University of Murcia, Murcia, Spain {pedro.mercader,abanos}@um.es

[2]Kristian Soltesz is with the Department of Automatic Control, Lund University, Lund, Sweden kristian@control.lth.se

The work has been supported in part by FEDER and Ministerio de Economía e Innovación under project DPI2013-47100-C2-1-P, and in part by the Swedish Research Council through the LCCC and ELLIIT centra. The first author is also supported by an FPU grant (FPU12/01026).

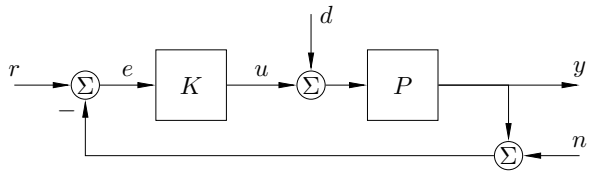

Fig. 1: Feedback control system, consisting of process P and compensator K. Signals are: process output y, compensator output u, load disturbance d and output disturbance (noise) n.

The paper is organized as follows: The problem is formulated in Section II, test-point methods are reviewed in Section III. Section IV introduces a method for probabilistic verification of constraint fulfilment. Three realistic synthesis examples are performed and analyzed in Section V.

II. Problem Statement

Figure 1 shows the considered setting, with process model P and compensator K. The process is modeled by the arbitrary-order time-delayed transfer function

$$P(s,\mathbf{p}) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0} e^{-hs}, \quad (3)$$

where $\mathbf{p} = [b_m, b_{m-1}, \ldots, b_0, a_n, a_{n-1}, \ldots, a_0, h]^\top$ is the stochastic process model parameter vector. The joint probabilistic density function (PDF) of $\mathbf{p}$ will be denoted $f(\mathbf{p}) : \Delta \mapsto \mathbb{R}^{m+n+1}$, Δ being the support of the PDF.

A PID compensator described by the following transfer function is considered

$$C(s,\mathbf{c}) = k_p + \frac{k_i}{s} + k_d s, \quad (4)$$

where $\mathbf{c} = [k_p, k_i, k_d]^\top$ is a deterministic vector of its parameters. In order to ensure high-frequency roll-of [4], the second order low-pass filter

$$G(s,T_f) = \frac{1}{s^2T_f^2/2 + sT_f + 1}, \quad (5)$$

parametrized in T_f, is connected in series with C, forming

$$K(s,\mathbf{k}) = G(s,T_f)C(s,\mathbf{c}), \quad (6)$$

where $\mathbf{k} = [\mathbf{c}^\top, T_f]^\top$ is the (deterministic) vector of compensator parameters.

The synthesis problem is formulated as a constrained optimization problem, where the objective is to maximize performance, under constraints ensuring robustness and noise attenuation. In this paper, the objective is to maximize load disturbance rejection. Two common ways to achieve this, is through minimization of either the integrated error

$$\text{IE}(\mathbf{p},\mathbf{c}) = \int_0^\infty e(t,\mathbf{p},\mathbf{c})dt, \quad (7)$$

or the integrated absolute error

$$\text{IAE}(\mathbf{p},\mathbf{c}) = \int_0^\infty |e(t,\mathbf{p},\mathbf{c})|\,dt. \quad (8)$$

The error e in both (7) and (8) is caused by a unit load disturbance step d, see Figure 1, applied with the control system in an equilibrium state.

For well-damped systems, it holds that IE $\approx$ IAE. However, oscillatory systems, with consecutive zero-crossings in e can yield small IE. From a performance measure point, this is not desired. Despite this, there remains one reason to consider minimization of IE in favor of IAE. It was shown in [5] that minimization of IE is equivalent to maximization of the integral gain k_i in (4). That is, IE minimization results in a convex objective. Unfortunately this is not the case for the IAE.

Robustness can be ensured by constraining the $\mathcal{H}_\infty$-norm of the sensitivity function

$$S(s,\mathbf{p},\mathbf{k}) = \frac{1}{1 + P(s,\mathbf{p})K(s,\mathbf{k})}. \quad (9)$$

In order to attenuate measurement noise, an $\mathcal{H}_\infty$-constraint is imposed on the noise sensitivity $Q = -KS$, being the transfer function from measurement noise n to compensator output u, see Figure 1. It is also straightforward to add similar constraints on other transfer functions, such as the complementary sensitivity $T = 1 - S$ (although this has not been done in this paper, to keep the presentation simple).

The deterministic counterparts to IE or IAE minimization under $\mathcal{H}_\infty$-constraints on S and Q have been thoroughly studied. Early work focused on robustness constrained IE minimization was presented in [5], [6]. Efficient convex-optimization-based algorithms for solving the same problem were recently presented in [7]. Robustness constrained minimization of IAE has been investigated in [8]. In [9], efficient gradient-based algorithms for the same problem were proposed. An extension for simultaneous compensator and filter design was presented in [10].

By combining (2) of Section I, with the IAE objective (8), and an $\mathcal{H}_\infty$-constraint M_s on (9), as proposed in this section, we arrive at the stochastic optimization problem

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & \mathbb{E}[\text{IAE}(\mathbf{p},\mathbf{k})], \\ \text{subject to} \quad & \mathbb{E}[S(s,\mathbf{p},\mathbf{k})] + \alpha\sqrt{\mathbb{V}[S(s,\mathbf{p},\mathbf{k})]} \leq M_s. \end{aligned} \quad (10)$$

It is straightforward to introduce additional constraints on complementary sensitivity T and noise sensitivity Q. By replacing the objective with its IE counterpart, defined through (7), one arrives at a simpler problem with the deterministic objective $-J(\mathbf{k}) = k_i$.

III. Test-point methods

This section serves as an introduction to test-point methods. Two such methods, the unscented transform (UT) [11], [12], and the conjugate unscented transform (CUT) [13], [14], [15], are considered. Both the UT and CUT can be used to obtain approximate solutions to optimization problems on the form (2).

Consider an arbitrary (nonlinear) function $g(\mathbf{x})$, where

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top, \quad (11)$$

is a multivariate Gaussian. It is well-known that any Gaussian can be transformed into one with zero expectation and unitary covariance, through an affine transformation. This transformation can be applied to g, and consequently, it is sufficient to consider $\mathbf{x}$ with $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and $\mathbb{V}[\mathbf{x}] = I$.

Denote by f the PDF of this distribution. The expectation $\mathbb{E}[g(\mathbf{x})]$ is defined as

$$\mathbb{E}\left[g(\mathbf{x})\right] = \int_{\mathbb{R}^n} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{12}$$

Test-point methods approximate $\mathbb{E}[g(\mathbf{x})]$ as a weighted sum

$$\mathbb{E}\left[g(\mathbf{x})\right] \approx \sum_{i=1}^{N} w_i g(\mathbf{x}^{(i)}), \tag{13}$$

where

$$\mathbf{x}^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right]^\top, \tag{14}$$

are known as the test points. Note that MC can be considered a test point method with weights $\omega_i = 1/N$, and test points $\mathbf{x}^{(i)}$ randomly generated by the underlying PDF. What makes the UT, CUT, and related test-points methods interesting is a clever choice of *deterministic* test-points and weights, removing the requirement of many samples (large N). Using the Taylor series expansion of $g(\mathbf{x})$ about the expected value $\mathbf{x} = \mathbf{0}$, (12) can be rewritten

$$\mathbb{E}\left[g(\mathbf{x})\right] = \sum_{N_1=0}^{\infty} \cdots \sum_{N_n=0}^{\infty} \frac{\mathbb{E}\left[x_1^{N_1} \cdots x_n^{N_n}\right]}{N_1! \cdots N_n!} \frac{\partial^{N_1+\cdots+N_n} g}{\partial x_1^{N_1} \cdots \partial x_n^{N_n}}(\mathbf{0}). \tag{15}$$

Combining (15) with (13) yields (16), to be found at the top of the next page.

Equating (15) and (16) leads to a set of equations

$$\sum_{i=1}^{N} w_i \left((x_1^{(i)})^{N_1} \cdots (x_n^{(i)})^{N_n}\right) = \mathbb{E}\left[x_1^{N_1} \cdots x_n^{N_n}\right], \tag{17}$$

referred to as the moment constraint equations (MCE). The idea behind test-point methods is to choose test points and corresponding weights, to fulfill all MCEs for which $N_1 + \cdots + N_n \leq d$, where d is referred to as the order of the MCE. This allows for *exact* integration of monomials up to order d, and Taylor approximation of other functions through such monomials.

Due to the symmetry of the Gaussian PDF f, odd moments are 0. The even moments up to order $d = 6$ are

$$\begin{aligned} &\mathbb{E}\left[x_i^2\right] = 1, \quad \mathbb{E}\left[x_i^4\right] = 3, \quad \mathbb{E}\left[x_i^2 x_j^2\right] = 1, \\ &\mathbb{E}\left[x_i^6\right] = 15, \quad \mathbb{E}\left[x_i^4 x_j^2\right] = 3, \quad \mathbb{E}\left[x_i^2 x_j^2 x_k^2\right] = 1, \end{aligned} \tag{18}$$

for distinct $i, j, k \in \{1, 2, \ldots, n\}$.

A. Unscented transform

The unscented transform (UT) [11], [12] relies on the selection of $N = 2n + 1$ test-points, satisfying the MCEs (17) up to order $d = 3$. This leaves some degree of freedom in choosing the test points. For the UT, they are constrained to lie on the principal axes. The resulting test points and corresponding weights are

$$\begin{aligned} \mathbf{x}^{(0)} &= \mathbf{0}, & w_0 &= \frac{\kappa}{n+\kappa}, \\ \mathbf{x}^{(i)} &= \sqrt{n+\kappa}\mathbf{e}_i, & w_i &= \frac{1}{2(n+\kappa)}, \\ \mathbf{x}^{(i+n)} &= -\sqrt{n+\kappa}\mathbf{e}_i, & w_{i+n} &= \frac{1}{2(n+\kappa)}, \end{aligned} \tag{19}$$

where $i = 1, 2, \ldots, n$, $\mathbf{e}_i$ is the unit vector along the i^{th} principal axis, and κ is a tuning parameter. For Gaussian $\mathbf{x}$, the choice

$$n + \kappa = 3 \tag{20}$$

was recommended by Julier and Uhlmann [12]. Adopting this recommendation, the test points and corresponding weights of (19) satisfy

$$\begin{aligned} &\textstyle\sum_{m=1}^{N} w_m = 1, \\ &\textstyle\sum_{m=1}^{N} w_m \left(x_i^{(m)}\right)^2 = 1 = \mathbb{E}\left[x_i^2\right], \\ &\textstyle\sum_{m=1}^{N} w_m \left(x_i^{(m)}\right)^4 = n + \kappa = 3 = \mathbb{E}\left[x_i^4\right], \\ &\textstyle\sum_{m=1}^{N} w_m \left(x_i^{(m)}\right)^2 \left(x_j^{(m)}\right)^2 = 0 \neq \mathbb{E}\left[x_i^2 x_j^2\right] = 1, \end{aligned} \tag{21}$$

for distinct $i, j \in \{1, 2, \ldots, n\}$. With the choice (20), the UT fulfills one of the 4^{th} order MCEs through $\mathbb{E}[x_i^4] = 3$. However, due to the fact that all UT test points lie along the principal axes, all cross moments are 0, as can be seen in the last equation of (21). Furthermore, the weight w_0 corresponding to the central point is negative for $n > 3$, leading to higher quadrature error compared to an equivalent method with positive weights [16]. These two aspects motivate the introduction of additional test points, not lying along the principal axes.

B. Conjugate unscented transform

The conjugate unscented transform (CUT) [13], [14], [15] proposes an extension of the UT test point set, by adding test points along conjugate coordinate axes. The conjugate-m axes lie in the directions of the vectors $\mathbf{c}_m^{(i)}$, generated by

$$\left\{\mathbf{c}_m^{(i)}, 1 \leq i \leq 2n\binom{n}{m}\right\} = \text{FS}\left[[\underbrace{1, \ldots 1}_{m}, \underbrace{0, \ldots, 0}_{m-n}]^\top\right]. \tag{22}$$

The $\text{FS}[\cdot]$ operator generates a fully symmetric set, closed under all sign and coordinate permutations. For instance, the unit vectors $\mathbf{e}_i$ along the principal axes together with their negated counterparts $-\mathbf{e}_i$, used in the UT, are generated by

$$\left\{\mathbf{s}^{(i)}, 1 \leq i \leq 2n\right\} = \text{FS}\left[[1, 0, \ldots, 0]^\top\right]. \tag{23}$$

Including test points along conjugate coordinate axes enables solving the MCEs up to an arbitrary order d, while maintaining positive weights. This comes at the cost of additional

$$\mathbb{E}\left[g(\mathbf{x})\right] \approx \sum_{N_1=0}^{\infty} \cdots \sum_{N_n=0}^{\infty} \frac{\sum_{i=1}^{N} w_i \left((x_1^{(i)})^{N_1} \cdots (x_n^{(i)})^{N_n} \right)}{N_1! \cdots N_n!} \frac{\partial^{N_1+\cdots+N_n} g}{\partial x_1^{N_1} \cdots \partial x_n^{N_n}}(\mathbf{0}). \tag{16}$$

TABLE I: Test-points for CUT4 (first three rows) and CUT6 (all rows).

Test points			Weights
Central	$\mathbf{x}^{(0)} = \mathbf{0}$		w_0
Principal	$\mathbf{x}^{(i)} = r_1 \mathbf{s}^{(i)}$	$1 \le i \le 2n$	w_1
Conjugate-n	$\mathbf{x}^{(i+2n)} = r_2 \mathbf{c}_n^{(i)}$	$1 \le i \le 2^n$	w_2
Conjugate-2	$\mathbf{x}^{(i+2n+2^n)} = r_3 \mathbf{c}_2^{(i)}$	$1 \le i \le 2n(n-1)$	w_3

test points. To illustrate, we will consider the CUT4 method, which fulfills all MCEs up to order $d = 4$ (actually $d = 5$). It has $N = 1 + 2n + 2^n$ test points:

- 1 central test point $\mathbf{x}^{(0)} = \mathbf{0}$ with weight w_0,
- $2n$ principal test points $\mathbf{x}^{(i)} = r_1 \mathbf{s}^{(i)}$ with weight w_1,
- 2^n conjugate-n test points $r_2 \mathbf{c}_n^{(i)}$ with weight w_2.

The scaling factors r_1, r_2, and weights w_1, w_2 for $n > 2$ utilized by CUT4 are given by

$$\begin{aligned} r_1 &= \sqrt{\frac{n+2}{2}}, \quad r_2 = \sqrt{\frac{n+2}{n-2}}, \\ w_1 &= \frac{4}{(n+2)^2}, \quad w_2 = \frac{(n-2)^2}{2^n (n+2)^2}, \quad w_0 = 0. \end{aligned} \tag{24}$$

For $n \le 2$ numerical values are presented in [13].

The CUT4 method corresponds to the first three rows of Table I. It is evident from the table that CUT4 is an extension of the UT (with different weights and scaling factors).

By adding $2n(n-1)$ conjugate-2 test points, one arrives at the CUT6 method, corresponding to rows 1–4 of Table I. Numeric values of corresponding weights and scaling factors can be found in [13]. The same work also presents parameters values for the CUT8 method.

IV. Probabilistic verification

Section III has provided tools, namely the UT and the CUT, to approximate stochastic optimization problems of the form (2), such as (10), by deterministic ones. It is of particular interest to investigate how such approximations affect the probability by which the resulting design violates constraints.

In the notation of (2), the probability of violating the i^{th} constraint is $1 - p_{\varphi_i}$, where p_{φ_i} is the constraint fulfillment probability

$$p_{\varphi_i} = \Pr\left(\varphi_i(s, \mathbf{k}, \mathbf{p}) \le 0\right). \tag{25}$$

There exists a framework known as randomized algorithms (RA), see for example [17], [18], which can be used to provide estimates $\hat{p}_{\varphi_i}$ of p_{φ_i}, lying within a priori specified accuracy $\epsilon \in (0, 1)$, with probability $1 - \delta$. This is formally states as

$$\Pr\left(|p_{\varphi_i} - \hat{p}_{\varphi_i}| \le \epsilon \right) \ge 1 - \delta. \tag{26}$$

A simple RA for constraint verification can be obtained through the MC method. The constraint fulfillment probability estimates $\hat{p}_{\varphi_i}$ are defined as

$$\hat{p}_{\varphi_i} = \frac{1}{M} \sum_{j=1}^{M} \mathcal{I}\left[\varphi_i(s, \mathbf{p}^{(j)}, \mathbf{k}) \le 0\right], \tag{27}$$

where $\mathcal{I}[\cdot]$ is 1 when its argument is true, and zero otherwise. The samples $\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(M)}$ are generated by the PDF f of $\mathbf{p}$. A lower bound on the number of samples required to meet (26) is given by the so-called Chernoff bound

$$\frac{1}{2\epsilon^2} \log \frac{2}{\delta} \le M. \tag{28}$$

The use of the RA, defined through (27) and (28), to compute posterior constraint violation probabilities, will be demonstrated in Section V-A.

V. Design Examples

This section is devoted to demonstrate the use of the introduced test-point methods for robust PID synthesis. Three realistic examples, highlighting different design aspects, will be considered.

In Example 1, the design of an ideal PI compensator for a first-order time-delayed process model is considered. The IE (7), defined in Section II, is minimized. Robustness is enforced through an $\mathcal{H}_\infty$-constraints on sensitivity. Feasible regions in compensator parameter space generated by UT, CUT4, and MC are compared. Confidence intervals for constraint violation of the CUT4 solution are computed using the RA of Section IV. These confidence intervals are compared to those obtained for a design disregarding parameter uncertainty (by assuming $\mathbf{p} = \mathbb{E}[\mathbf{p}]$). Example 2 illustrates the advantage of minimizing IAE (8), when the process model is weakly damped. Finally, the role of the constraint on noise sensitivity $Q = -KS$ is demonstrated in Example 3.

A. Example 1: First-order time-delayed process

Consider the process model

$$P(s, \mathbf{p}) = \frac{b}{s + a} e^{-sh}, \tag{29}$$

where $\mathbf{p} = [b, a, h]^\top$ is a Gaussian, defined through $\mathbb{E}[\mathbf{p}] = [1, 1, 1]^\top$ and $\mathbb{V}[\mathbf{p}] = 0.05 I_3$. Synthesis of an ideal PI

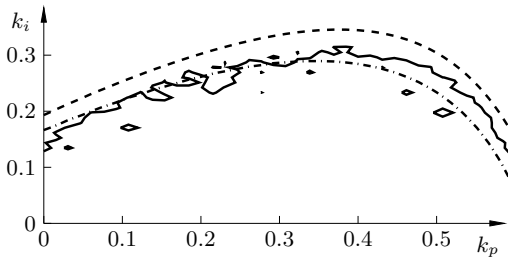

Fig. 2: Boundary of the feasible region in (k_p, k_i)-space corresponding to (30), for MC with 10^4 samples (solid), UT (dashed line), and CUT4 (dash-dot line).

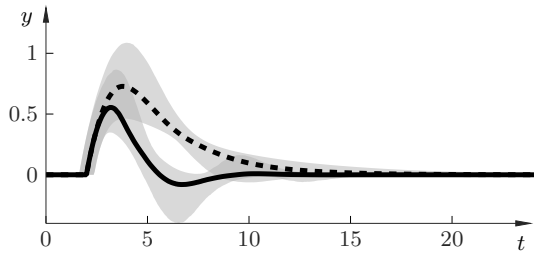

Fig. 3: Median load disturbance step responses for $\mathbf{c}_{\text{opt}}$ (solid) and $\mathbf{c}_{\text{nom}}$ (dashed). Grey areas indicate the 5–95 % quantiles over 10^4 MC simulations.

compensator $C(s, \mathbf{c}) = k_p + k_i/s$ with $\mathbf{c} = [k_p, k_i]^\top$ is investigated. Due to the lack of oscillatory modes, minimization of the expected IE is performed (through maximization of k_i). Robustness is ensured by imposing the (industrially relevant) constraint level $M_s = 1.6$. The resulting optimization problem is given below.

$$\begin{aligned} &\underset{\mathbf{c}}{\text{minimize}} \;\; -k_i, \\ &\text{subject to} \\ &\mathbb{E}\left[\|S(s,\mathbf{p},\mathbf{c})\|_\infty\right] + \sqrt{\mathbb{V}\left[\|S(s,\mathbf{p},\mathbf{c})\|_\infty\right]} \leq 1.6. \end{aligned} \tag{30}$$

Feasibility boundaries in (k_p, k_i)-space, obtained as contour plots corresponding to $M_s = 1.6$, are shown in Figure 2 for UT, CUT4, and MC. The MC, using 10^4 samples, results in a non-compact feasible region with uneven boundary. Doubling the number of sample points improved the situation only marginally, indicating the need for a significant increase in samples. However, already with 10^4 samples, the MC takes 15 min to execute.

The execution time of each method is approximately proportional to the number of samples or test points. The CUT4 method (with 14 test points) executed 720 times faster than MC (with 10^4 samples). UT (with 6 test points) was twice as fast as CUT4. The actual execution time for each method is machine-dependent, but the CUT4 version executes in < 1 s on a standard computer. Consequently, we recommend the use of CUT4, as it executes sufficiently fast, while having the potential for better approximation than the UT.

The CUT4 solution of (30) is $\mathbf{c}_{\text{opt}} = [0.34, 0.29]^\top$. Figure 3 compares load disturbance attenuation of this solution to that of $\mathbf{c}_{\text{nom}} = [0.46, 0.51]^\top$, obtained when the uncertainty is ignored (by assuming $\mathbf{p} = \mathbb{E}[\mathbf{p}]$).

Using the RA of Section IV, we find a high confidence for the probability of constraint fulfillment with $\mathbf{c}_{\text{opt}}$ to be 93 %:

$$\Pr\left(\left|\Pr\left(\|S(s,\mathbf{p},\mathbf{c}_{\text{opt}})\|_\infty \leq 1.6\right) - 0.93\right| \leq 0.02\right) \geq 0.98.$$

However, the solution $\mathbf{c}_{\text{nom}}$ obtained without considering the uncertainty of $\mathbf{p}$ can only provide equal confidence for the much lower probability of 51 %. Furthermore, there is a significant probability of 18 % for $\mathbf{c}_{\text{nom}}$ violating $M_s = 2.0$ (corresponding to poor closed-loop robustness):

$$\Pr\left(\left|\Pr\left(\|S(s,\mathbf{p},\mathbf{c}_{\text{nom}})\|_\infty \leq 2.0\right) - 0.82\right| \leq 0.02\right) \geq 0.98.$$

The effect of neglecting the stochastic nature of $\mathbf{p}$ gets even worse for this example when PID (as opposed to PI) compensator design is considered.

B. Example 2: Oscillatory process

Consider the process model

$$P(s,\mathbf{p}) = \frac{b}{s^2 + s + a} e^{-sh}, \tag{31}$$

where $\mathbf{p} = [b, a, h]^\top$ is a multivariate Gaussian with expected value $\mathbb{E}[\mathbf{p}] = [5, 5, 0.1]^\top$ and variance

$$\mathbb{V}[\mathbf{p}] = \begin{bmatrix} 0.25 & 0.1 & 0 \\ 0.1 & 0.25 & 0 \\ 0 & 0 & 10^{-5} \end{bmatrix}. \tag{32}$$

The nominal process model has a strongly oscillative mode, with relative ratio $\zeta = 0.22$.

Using CUT4 to design an ideal PID compensator on the form (4) results in $\mathbf{c}_{\text{IE}} = [1.38, 5.70, 0.81]^\top$, with objective and constraints according to (30) of Example 1. It is well-known that minimization of the IE is not suitable for oscillatory processes [8], for which minimization of the IAE is preferential. (However, the IE minimizing solution can be used to warm-start the IAE minimization.) Replacing the objective $-k_i$ of (30) for $\mathbb{E}[\text{IAE}(\mathbf{p},\mathbf{c})]$ yields $\mathbf{c}_{\text{IAE}} = [1.76, 3.46, 0.72]^\top$. A comparison of load step responses between $\mathbf{c}_{\text{IE}}$ and $\mathbf{c}_{\text{IAE}}$ is shown in Figure 4.

C. Example 3: Measurement noise filtering

We will anew consider the uncertain process model of Example 1. This time, the process model is subject to additive white measurement noise of zero mean and standard deviation $\sigma_n = 0.1$. Performing the design for an ideal PID compensator (4) using (30) results in $\mathbf{k}_\infty = [0.53, 0.47, 0.26, 0]^\top$, where the last zero corresponds to the filter $G(s, 0) = 1$, see (5). As shown in Figure 5b, $\mathbf{k}_\infty$ results in unacceptably high compensator output activity. Consequently, an additional constraint is imposed on the