Krist V. Gernaey, Jakob K. Huusom and Rafiqul Gani (Eds.), 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering.
31 May - 4 June 2015, Copenhagen, Denmark. 

# Improved Optimization-based Design of PID Controllers Using Exact Gradients

Chriss Grimholt and Sigurd Skogestad*

*Department of Chemical Engineering; NTNU; Trondheim, Norway*
**e-mail: skoge@ntnu.no*

## Abstract

Finding good controller settings that satisfy complex design criteria is not trivial, even for the simple three parameter *proportional-integral-derivative* (PID) controller. One strategy is to formulate the design problem into an optimization problem. However, when using gradient based optimization with finite differences to estimate the gradients, the algorithm often fails to converge to the optimal solution. This is thought to be a result of inaccuracies in the estimation of the gradients. In this paper we show exact gradients for a typical performance (IAE) versus robustness ($M_S$, $M_T$) design problem. We demonstrate increased accuracy in the optimization when using exact gradients compared with forward finite differences.

**Keywords:** PID control, controller design, optimization, exact gradients, IAE, $M_S$.

## 1. Introduction

The simple three parameter *proportional-integral-derivative* (PID) controller is the most adopted controller in the process industry. However, finding good parameter values by trial and error is not only difficult, but also time consuming. In combination with simple models, good parameters are usually found using tuning rules like Ziegler-Nichols or SIMC.

When the design complexity increases, in the form of process model complexity or special requirements on controller performance or robustness, it is beneficial to switch to optimization-based design. Our optimization problem can be stated as follows,

$$\begin{aligned} \underset{parmeters}{\text{minimize:}} \quad & \text{performance cost } (J) \\ \text{subject to:} \quad & \text{required robustness} \end{aligned}$$

The background for this study was to find optimal PID controller settings for a first-order plus delay process, and also to find optimal settings for a Smith Predictor controller. Here, the performance requirements were to minimize integrated absolute error (IAE) for input and output disturbances, and the robustness criterion was to have a given sensitivity peak ($M_S$). A similar design formulation has been proposed by Shinskey (1990) where load disturbance was optimized when subjected to sensitivity constraints. By using the link between integral error and integral gain (IE $= 1/k_i$), Åström et al. (1998) and Panagopoulos et al. (2002) formulated the optimization problem as a set of algebraic equations which could be efficiently solved. However, using IE as the performance criterion can lead to oscillatory response, especially for PID controllers.

Initially, we solved this optimization problem using gradient-free optimization similar to the work done by Garpinger and Hägglund (2008). Natively, gradient-free approaches, like the simplex method, does not explicitly handle constraints. To bypass this, the constraints are handled internal by an internal solver. For our application, generating optimal trade-off curves for PID controllers, requiring hundreds of accurate sequential optimizations, the gradient-free method was too slow due to the internal solver and the number of iterations required to converge. In addition, the optimal point was slightly inaccurate

By switching to gradient based methods, using finite-differences to estimate the gradients, we achieved faster convergence. However, the optimization algorithm frequently failed to converge to the solution. Surprisingly, this also happens even though the initial guess is very close to the local optimum. In our case, it seems that the main problem is not necessarily the non-convexity of the problem and the possibility for local minima, but rather inaccuracies in the estimation of the gradients when using finite-differences. We found that the robustness of the optimization was significantly improved by supplying the exact gradients.

In this paper we show and use exact gradients for IAE of the time response and the peak of the sensitivity functions $M_S$, and $M_T$. The main advantage of using accurate gradients is that we improve the convergence properties and make the problem less sensitive to the initial point. The approach has been successfully used to find optimal PID on a first-order with delay processes. But the method can also easily be extended to other processes and controllers.

## 2. The closed-loop system

In this paper we consider the linear feedback system as shown in Figure 1. Disturbances can enter the system at two different locations, at the plant input ($d_u$) and the plant output ($d_y$). The disturbance at the plant output is equivalent to a setpoint change. However, unlike setpoint changes, the output disturbance is not known in advance. Measurement noise ($n$) enters the system at the measured output ($y$). This system is represented by four transfer functions nicknamed *the gang of four*,

$$S(s) = \frac{1}{1 + G(s)\,K(s)} \qquad T(s) = 1 - S(s)$$

$$GS(s) = G(s)\,S(s) \qquad KS(s) = K(s)\,S(s)$$

where their effect on the control error and plant input is,

$$-e = S(s)\,d_y + GS(s)\,d_u + T(s)\,n \tag{1}$$

$$-u = KS(s)\,d_y + T(s)\,d_u + KS(s)\,n \tag{2}$$

Though we could use any fixed-order controller, we chose in this paper to use the parallel PID controller as an example;

$$K_{\text{PID}}(s;p) = k_p + \frac{k_i}{s} + k_d s = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right); \quad p = \begin{pmatrix} k_p & k_i & k_d \end{pmatrix}^T \tag{3}$$

where $k_p = K_c$, $k_i = K_c/\tau_I$, and $k_d = K_c\,\tau_D$ is the proportional, integral, and derivative gain, respectively. The parallel PID controller can have complex zeros, which we have observed can result in several peaks or plateaux for the magnitude of sensitivity function in the frequency domain $|S(j\omega)|$. This becomes important when adding specifications on the frequency behaviour, which we will show later.

# 3. Quantifying the optimal controller

### *3.1. Performance*

In this paper we chose one of the most popular ways of quantifying controller performance, the integrated absolute error (IAE);

$$\mathrm{IAE}\,(p) = \int_0^{t_f} |e(t;\,p)|\ \mathrm{d}t \tag{4}$$

when subjecting the system to a disturbance. We take both input and output disturbances into account and choose the weighted cost function,

$$J(p) = 0.5\left(\varphi_{dy}\ \mathrm{IAE}_{dy}(p) + \varphi_{du}\ \mathrm{IAE}_{du}(p)\right); \quad \varphi_{du} = \frac{1}{\mathrm{IAE}^{\circ}_{du}}; \quad \varphi_{dy} = \frac{1}{\mathrm{IAE}^{\circ}_{dy}} \tag{5}$$

where both terms are weighted equally with 0.5 to get a good balance. The $\varphi_{dy}$ and $\varphi_{du}$ are scaling factors from IAE-optimal PID controllers for a step load change on the input and output, respectively. To ensure robust reference controllers, they are required to have $M_S = M_T = 1.59$. Note that two different controllers are used to obtain the reference $\mathrm{IAE}^{\circ}$ values, whereas a single controller $K(s;p)$ is used to find $\mathrm{IAE}_{dy}(p)$ and $\mathrm{IAE}_{du}(p)$.

### *3.2. Robustness*

In this paper we have choosen to quantify robustness in terms of $M_S$ and $M_T$;

$$M_S = \max_{\omega} |S(j\omega)| = \|S(j\omega)\|_{\infty}$$
$$M_T = \max_{\omega} |T(j\omega)| = \|T(j\omega)\|_{\infty}$$

where $\|\cdot\|_{\infty}$ is the $H_{\infty}$-norm. In the Nyquist plot, $M_S$ is the inverse of the closest distance between the critical point $(-1, 0)$ and the loop transfer function $L(s) = G(s)\,K(s)$.

The closed-loop system is insensitive to small process variations for frequencies where the sensitivity function $\left|S(j\omega)\right|$ is small (Åström and Hägglund, 2006). On the other hand, small values of the complementary sensitivity function $|T(j\omega)|$ tells us that the closed-loop system is insensitive to larger process variation. The system is most sensitive at the peak of the sensitivity functions ($M_S$ and $M_T$).

From experience, using only the peaks can lead to cycling between iterations. This is typical for PID control on first order models, where the optimal controller has several peaks of equal magnitude in $S(j\omega)$. The optimizer then discreetly jumps from using one peak to the other between iterations. It is assumed that this results from the peaks having different gradients. To avoid this problem, instead of putting the upper bounds on the peaks, e.g.

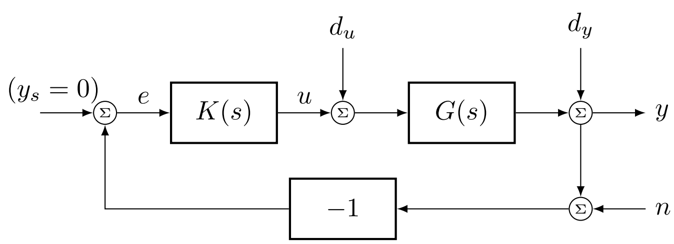


Figure 1: Block diagram of the closed loop system, with controller $K(s)$ and plant $G(s)$.

$M_S \leq M_S^{ub}$, we choose to require

$$\left|S(j\omega)\right| \leq M_S^{ub} \qquad \text{for all } \omega$$

which in addition to handling multiple peaks and plateaux, also improves convergence for infeasible initial controllers. The constrain is approximated by gridding the frequency domain within the interesting region, resulting an inequality constraint for each grid point;

$$\left|S(j\omega)\right| \leq M_S^{ub} \qquad \text{for all } \omega \text{ in } \Omega$$

where $\Omega$ is the set of selected frequency points during the gridding of the frequency domain. This results in reduced accuracy and increased computational load. However, the benefit gained towards improved convergence properties makes up for this.

## 4. Implementation of the optimization problem

The optimization problem can be stated as follows,

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & J(p) = 0.5\left(\varphi_{dy}\,\text{IAE}_{dy}(p) + \varphi_{du}\,\text{IAE}_{du}(p)\right) \\ \text{subject to} \quad & c_s(p) = |S(j\omega;\,p)| - M_S^{ub} \leq 0 \quad \text{for all } \omega \text{ in } \Omega \\ & c_t(p) = |T(j\omega;\,p)| - M_T^{ub} \leq 0 \quad \text{for all } \omega \text{ in } \Omega \end{aligned}$$

where $M_S^{ub}$ and $M_T^{ub}$ are the upper bounds on $\left|S(j\omega)\right|$ and $\left|T(j\omega)\right|$. If there is a trade-off between performance and robustness, at least one robustness constraint will be active.

## 5. Gradients

The gradient of a function $\nabla f(p)$ with respects to $p$ is defined as

$$\nabla_p f(p) = \begin{pmatrix} \frac{\partial f}{\partial p_1} & \frac{\partial f}{\partial p_2} & \cdots \frac{\partial f}{\partial p_n} \end{pmatrix}^T \tag{6}$$

where $n_p$ is the number of parameters. In this paper, $p_i$ reference to the parameter $i$, and the partial derivative $\frac{\partial f}{\partial p_i}$ is called the sensitivity of $f$. The sensitivities can be approximated by forward finite differences,

$$\frac{\partial f}{\partial p_i} \approx \frac{f(p_i + \Delta p_i) - f(p_i)}{\Delta p_i} \tag{7}$$

requiring $(1 + n_p)$ perturbations. For our problem this results in $2(1 + n_p)$ step response simulations. The accuracy can be improved by using central differences, which requires $(1 + 2n_p)$ perturbations.

### *5.1. Cost function gradient*

The gradient of the cost function $\nabla_p J(p)$ is then expressed in terms of the sensitivities,

$$\frac{\partial J(p)}{\partial p_i} = 0.5\left(\varphi_{dy}\,\frac{\partial \text{IAE}_{dy}(p)}{\partial p_i} + \varphi_{du}\,\frac{\partial \text{IAE}_{du}(p)}{\partial p_i}\right) \tag{8}$$

For even relatively simple systems with time delay, these integrals becomes quite complicated, if not impossible, to solve. However, the sensitivities can be found in a fairly straight forward manner by developing them such that the integrals can be numerically evaluated. The sensitivity of IAE can be expressed as,

$$\frac{\partial \text{IAE}(p)}{\partial p_i} = \int_0^{t_f} \text{sign}\left\{e(t;p)\right\}\left(\frac{\partial e(t;p)}{\partial p_i}\right) \mathrm{d}t \tag{9}$$

where $e(t;p) = \mathcal{L}^{-1}\left\{e(s;p)\right\}$ and $\frac{\partial e(t;p)}{\partial p_i} = \mathcal{L}^{-1}\left\{\frac{\partial e(s;p)}{\partial p_i}\right\}$

For the case given in Figure 1 with unit steps load disturbances, the error $e(s;p)$ is ${S(s;p)}/{s}$ for the output disturbance $dy$ and ${GS(s;p)}/{s}$ for the input disturbance $du$. The transient ${\partial\, e(t;p)}/{\partial p}$ can then easily found by step response simulations of

$$\frac{\partial S(s)}{\partial p_i} = -GS(s)\; S(s)\; \frac{\partial K(s)}{\partial p_i} \tag{10}$$

$$\frac{\partial GS(s)}{\partial p_i} = -GS(s)\; GS(s)\; \frac{\partial K(s)}{\partial p_i} \tag{11}$$

The gradient of IAE is then calculated by evaluating (9) using numerical integration, and the method is closely related to using direct sensitivities in optimal control problems. This results in $2(1+n_p)$ step response simulations, which is the same as the forward finite differences approximation in (7).

### *5.2. Constraint gradients*

The gradient of the robustness constraints $\nabla c_s(p)$ and $\nabla c_t(p)$ expressed in terms of the parameter sensitivities are,

$$\frac{\partial c_s(j\omega)}{\partial p_i} = \frac{\partial |S(j\omega)|}{\partial p_i} = \frac{1}{\left|S(j\omega)\right|}\,\mathrm{Re}\left\{S(-j\omega)\,\frac{\partial S(j\omega)}{\partial p}\right\} \qquad \text{for all } \omega \text{ in } \Omega \tag{12}$$

$$\frac{\partial c_t(j\omega)}{\partial p_i} = \frac{\partial |T(j\omega)|}{\partial p_i} = \frac{1}{\left|T(j\omega)\right|}\,\mathrm{Re}\left\{T(-j\omega)\,\frac{\partial T(j\omega)}{\partial p}\right\} \qquad \text{for all } \omega \text{ in } \Omega \tag{13}$$

If the constrains were instead expressed in terms of the peaks of the transfer functions (e.g. $M_S \le M_S^{ub}$) the gradients become as follows,

$$\frac{\mathrm{d}}{\mathrm{d}p_i}\|G(j\omega;p)\|_\infty = \frac{\partial}{\partial p_i}|G(j\omega_{peak};p)| \tag{14}$$

where $\omega_{peak}$ is the frequency of the peak.

## 6. Case study

The exact gradients were implemented for the problem;

$$G(s) = \frac{e^{-s}}{s+1};\ \mathrm{IAE}^\circ_{dy} = 1.56;\ \mathrm{IAE}^\circ_{du} = 1.42;\ M_S^{ub} = M_T^{ub} = 1.3;\ p_0 = \begin{pmatrix} 0.2 \\ 0.02 \\ 0.3 \end{pmatrix} \tag{15}$$

To make the loop function proper, a first order filter with filter time constant $\tau_F = 0.001$ was added to the controller. The error response and parameter sensitivity was found by fixed step integration ($t_f = 25$, $n_{\mathrm{steps}} = 10^4$), with the initial point $p_0$. The problem was solved using Matlab's fmincon with the active set algorithm. For comparison, forward finite differences was used to approximate the gradients. This problem has two equal $M_S$ peaks at the optimum (Figure 2), and is a typical example of a problem exhibiting cyclic behaviour when using the $M_S \le M_S^{ub}$ constraint. The optimal error response is shown in Figure 3.

The exact gradients performed better than the finite differences (Table 1). The exact cost function gradients give the best improvement. This signals that the optimum is relatively flat, and that the approximated cost function gradients are not precise enough to find the true local optimum. The same test was performed for different numbers of time steps during the integration. Even with $n_{\mathrm{steps}} = 10^5$, the forward finite differences failed to converge to the local optimum (controller parameter error in the second digit). On the other hand,

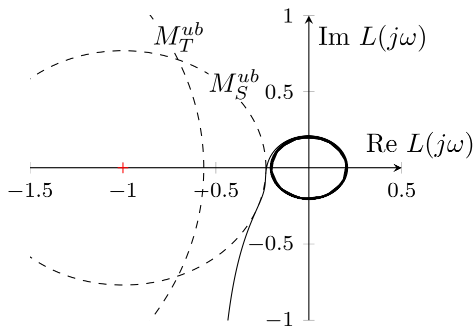


Figure 2: Nyquist plot of $L(j\omega)$ with the optimal controller for the problem given in (15).

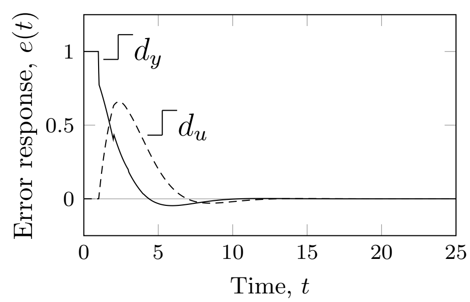


Figure 3: Optimal error response for the problem given in (15).

the exact gradient could still converge to the local optimum with as low as $n_{\text{steps}} = 500$ (control parameters error in the fifth digit). The exact gradient converged for most stable initial guesses, but the forward finite differences failed to find the optimum even when started very close to the optimum $p_0 = \begin{pmatrix} 1.001p_1^\star & p_2^\star & p_3^\star \end{pmatrix}^T$. When using central finite differences, the accuracy was increased. However, this requires $2(1+2n_p)$ step simulations.

Table 1: Comparison between optimal solutions with different combinations of gradients.

| Gradient type | | Cost function | Optimal parameters | | | number of |
|---|---|---|---|---|---|---|
| Cost-function | Constraints | $J(p^\star)$ | $k_p$ | $k_i$ | $k_d$ | iterations |
| exact | exact | 2.0598 | 0.5227 | 0.5327 | 0.2172 | 13 |
| fin.dif. | exact | 2.1400 | 0.5204 | 0.4852 | 0.1812 | 16 |
| exact | fin.dif. | 2.0598 | 0.5227 | 0.5327 | 0.2172 | 13 |
| fin.dif. | fin.dif. | 2.9274 | 0.3018 | 0.3644 | 0.2312 | 11 |

## 7. Conclusion

In this paper we have successfully applied the exact gradients for a typical performance (IAE) versus robustness ($M_S$, $M_T$) optimization problem. The exact gradients improved the convergence to the true optimal point when compared to gradients approximated by forward finite difference.

## References


Åström, K., Panagopoulos, H., Hägglund, T., May 1998. Design of PI Controllers based on Non-Convex Optimization. Automatica 34 (5), 585–601.

Åström, K. J., Hägglund, T., 2006. Advanced PID control. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709.

Garpinger, O., Hägglund, T., 2008. A software tool for robust pid design. In: Proc. 17th IFAC World Congress, Seoul, Korea.

Panagopoulos, H., Å ström, K., Hägglund, T., 2002. Design of PID controllers based on constrained optimisation. IEE Proceedings - Control Theory and Applications 149 (1), 32.

Shinskey, F. G., 1990. How Good are our controllers in absolute performance and robustness. Measurement and Control 23, 114–120.