Multi-loop Decentralized PID Control Based on Covariance Control Criteria: an LMI Approach

Xin Huang\(^1\), Biao Huang\(^1\)\(^*\)

\(^1\)Department of Chemical & Materials Engineering
University of Alberta
Edmonton, Alberta, Canada T6G2G6

Abstract

PID control is well known and widely applied in industry and many design algorithms are readily available in the literature. However, systematic design of multi-loop or decentralized PID control for multivariable processes to meet certain objectives simultaneously is still a challenging task. Designing multi-loop PID controllers such that the process variables satisfy the generalized covariance constraints is studied in this paper. A convergent computational algorithm is proposed to calculate the multi-loop PID controller for a process with stable disturbances. This algorithm is then extended to a process with random-walk disturbances. The feasibility of the proposed algorithm is verified by applying it to several simulation examples.

Keywords: PID; Covariance control; LMI; Semidefinite programming;

\(^*\)Corresponding author. Email Biao.Huang@ualberta.ca, tel:(780)492-9016, Fax:(780)492-2881
1 Introduction

The proportional-integral-derivative (PID) controller is extensively used in industry and is well documented in the literature since the classic Ziegler-Nichols method [27] was presented. This is because PID controller is simple, robust and well understood. To accommodate the high performance requirement of the modern industry, optimization of the PID parameters is also extensively studied, and many different tuning criteria and procedures have been proposed, for example, decay ratio method [5], gain and phase margin method [2] and the internal model control (IMC) based PID tuning method [17, 21]. Recently, with the popularity of the interior point algorithm several PID design methods based on Linear Matrix Inequality (LMI) were proposed for the continuous-time systems [3, 7, 9]. However, none of the above mentioned PID design methods are presented to directly achieve variance specification on all outputs for multivariable systems.

The significance of reducing the process variation is well appreciated in the manufacture industry [23]; however, there are only a few papers on optimizing PID parameters in order to reduce the process variances directly. A stochastic predictive PID controller was presented in [13, 18] by equating a discrete PID control law with the linear form of the Generalized Predictive Control (GPC) with steady state weighting; several self-tuning PID controllers were proposed in [16, 22, 26] for discrete-time Linear Time-invariant (LTI) systems by approximating the PID controller to the generalized minimum variance control (GMVC). The philosophy behind these methods [13, 18, 16, 22, 26] is to design the PID parameters by approximation of a PID controller to a controller designed by other advanced methods. In such a way, it is expected that the PID controller has similar property as the other advanced controllers. However, there is no theoretical guarantee about how “close” such approximation can be; furthermore, these approximation methods so far are available only for the single-input-single-output (SISO) systems and the extension of the PID design from the SISO system to the multi-input-multi-output (MIMO) system is nontrivial.

In this paper a state space approach to designing multi-loop PID controllers is proposed such that closed-loop satisfies the generalized covariance constraints. One of the main advantages of the proposed method is that the controller parameters are calculated directly according to the covariance constraints on process variables instead of approximating other controllers. A convergent computational algorithm, in which a sequence of semi-definite programming problems are solved using the LMItool [8], is proposed to calculate the multi-loop PID controller parameters. The proposed algorithm initially intends for the process with stable disturbances. The algorithm is then extended to the process with unstable disturbance (random walk disturbance). The proposed multi-loop PID controller design method is for the purpose of controlling the variation of process variables. However, the proposed approach can also be applied to design multi-loop PID controller for other performance indices, such as $H_2$ or $H_\infty$.

The rest of the paper is organized as follows: the generalized covariance constraints problem and the preliminary results are stated in Section 2. The state space realization of the multi-loop PID controller is given in Section 3. An algorithm is presented to calculate the multi-loop PID
controller parameters in Section 4, where the disturbance model is assumed to be stable. The multi-loop PID controller design for the random walk disturbance is addressed in Section 5. Numerical examples are presented in Section 6 followed by concluding remarks in Section 7.

The notation of this paper is quite standard: \( \mathbb{R} \) denotes the set of real number; \( \mathbb{R}^n \) denotes the \( n \)-dimensional real vector space; \( I_n \) means the unit element in linear space \( \mathbb{R}^{n \times n} \); \( A^T \) stands for the transpose of matrix \( A \); \( A^{-1} \) means inverse of matrix \( A \); \( X > Y \) (\( X \geq Y \)) means that \( X - Y \) is positive definite (semi-definite); \( E \) is the expectation operator; \( \text{var}(y) \) denotes variance of \( y \); \( \delta (k) \) is the discrete-time Dirac function: 
\[
\delta (k) = \begin{cases} 
1 & k = 0 \\
0 & k \neq 0 
\end{cases};
\]
\( z^{-1} \) is the back shift operator: \( z^{-1}y_k = y_{k-1} \); \( \Delta = 1 - z^{-1} \) is the difference operator: \( \Delta y_k = y_k - y_{k-1} \).

2 Problem statement

Consider a finite-dimensional discrete-time LTI system \( \mathcal{P} \), given as follows:
\[
\begin{align*}
  x_{k+1} &= Ax_k + Bu_k + G\zeta_k \\
  y_k &= Cx_k + F\zeta_k \\
  z^i_k &= C_i x_k + D_i u_k, \ i = 1...l
\end{align*}
\]
(1)

where \( x_k \in \mathbb{R}^n \) is the state vector, \( u_k \in \mathbb{R}^m \) is the input, \( y_k \in \mathbb{R}^m \) is the measurement, \( \zeta_k \in \mathbb{R}^g \) is the external disturbance and measurement noise, and \( z^i_k \in \mathbb{R}^{p_i} \) is the \( i \)-th controlled vector. \( A, B, G, C, F, C_i \) and \( D_i \) are matrices with appropriate dimensions. The disturbance \( \zeta_k \) is unmeasurable, but it is assumed that some of its statistical properties are known:
\[
\begin{align*}
  E(\zeta_k) &= 0 \\
  E(\zeta_i^{T} \zeta_j) &= \Omega \delta (i - j)
\end{align*}
\]
(2)

It is assumed that the state space representation is a minimal realization.

To achieve better product quality, controlling the variation of the process variables is well accepted in industry [23], and different control strategies have been presented, such as minimum variance control [1], the generalized minimum variance control (GMVC) [26], linear quadratic Gaussian (LQG) control [12, 14, 15] and the covariance control [24, 25]. For multivariable systems controlling the plant’s covariance is one of the main objectives [24, 25]. The covariance of \( x_k \) is defined as:
\[
\Sigma = \lim_{k \to \infty} E \left( x_k x_k^T \right)
\]
However, as it is pointed out in [10], there is often no physical interpretations of the covariance of the states. Therefore, it is more desirable to control covariance of process output rather than that of states. The generalized covariance constrained control (GCC) problem is stated as follows:

**Problem 2.1**: For the continuous-time LTI system (1), find a controller such that the closed-loop system is internally stable and the covariance of the controlled variable \( z^i_k (i = 1...l) \) satisfies
\[
\Phi_i = \lim_{k \to \infty} E \left( z^i_k z_k^T \right) < \Phi_{i0}
\]
(4)
where \( \Phi_i (i = 1 \ldots l) \) is some pre-specified positive definite matrix.

If there exists a multi-loop PID controller such that the closed-loop system is internally stable and the generalized covariance constraints (4) are satisfied, then the GCC problem is feasible via a multi-loop PID controller. It has been shown [10] that the feasibility of GCC is equivalent to feasibility of some linear matrix inequalities (LMIs) if a full order dynamic controller is considered. Unfortunately, this conclusion does not hold for the fixed-order controller and decentralized controller. The interior point algorithm can be used to solve LMI problems quite efficiently [4, 8], but it can not be used here to solve the multi-loop PID design with the generalized covariance constraints. However, an iterative algorithm, in which a sequence of optimization problems are solved, is proposed in section 5 to calculate the multi-loop PID controller parameters. To state the computational algorithm, the well known Schur complement Lemma will be used:

**Lemma 2.1** (Schur complement lemma) The following statements are equivalent:

1) \[
\begin{bmatrix}
  A & B \\
  B^T & C
\end{bmatrix} > 0
\] (5)

2) \( C > 0 \) and \( A - BC^{-1}B^T > 0 \).

### 3 Multi-loop PID controller

Multi-loop PID controllers have certain advantages over the complex multivariable control systems. Multi-loop PID controllers are easier to implement on DCS and requires less training compared to the multivariable controller. However, optimization of the PID parameters to reduce the effect of disturbance is not a trivial task due to the non-convex nature of the optimization problem. The presented approach is based on the state space representation and a state realization of the multi-loop PID controller is given in this section.

A discrete-time single-loop PID controller can be described as:

\[
 u_k = k_1 e_k + k_2 \sum_{i=0}^{k} e_i + k_3 (e_k - e_{k-1})
\] (6)

where \( u_k \) is the manipulated variable and \( e_k = r_k - y_k \) is the error between setpoint \( r_k \) and the measurement \( y_k \). The velocity form of the discrete-time PID controller can be obtained from equation (6):

\[
 \Delta u_k = (k_1 + k_2 + k_3) e_k + (-k_1 - 2k_3) e_{k-1} + k_3 e_{k-2}
\] (7)

The transfer function of the discrete-time PID controller (6) can be obtained easily from the velocity form:

\[
 C(z^{-1}) = \frac{(k_1+k_2+k_3)+(-k_1-2k_3)z^{-1}+k_3z^{-2}}{1-z^{-1}}
\] (8)
The controllable state space realization of the PID controller (6) is obtained from the transfer function (8):

\[ x_{k+1}^s = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} x_k^s + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e_k \]

\[ u_k^s = \begin{bmatrix} \tilde{k}_3 & \tilde{k}_2 \end{bmatrix} x_k^s + \tilde{k}_1 e_k \]  

where \( \tilde{k}_1 = k_1 + k_2 + k_3, \tilde{k}_2 = k_2 - k_3, \tilde{k}_3 = k_3 \) and \( x_k^s \) represents the state vector of a single-loop PID. For the multivariable systems, the multi-loop PID controller \( C_m(z^{-1}) \), consisted of a group of individual PID controllers, is given as:

\[ C \left( z^{-1} \right) = \text{diag} \left( c_1 \left( z^{-1} \right), c_2 \left( z^{-1} \right), \ldots, c_m \left( z^{-1} \right) \right) \]  

where \( c_i \left( z^{-1} \right) \) is the \( i \)th single-loop PID controller with the same form as (8). With this multi-loop PID controller, the diagram of the closed-loop system is shown as:

![Figure 1: Closed-loop diagram](image)

The state space representation of the multi-loop PID controller can be obtained by stacking the state space realization of each individual PID controller (where superscript \( m \) represents multi-loop):

\[ x_{k+1}^m = A_c x_k^m + B_c e_k^m \]

\[ u_k^m = C_c x_k^m + D_c e_k^m \]  

where matrices \( A_c, B_c, C_c \) and \( D_c \) are defined as:

\[ A_c = \text{diag} \left( \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right) \in R^{2m \times 2m} \]

\[ B_c = \text{diag} \left( \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \in R^{2m \times m} \]

\[ C_c = \text{diag} \left( \begin{bmatrix} \tilde{k}_3^m \\ \tilde{k}_2^m \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{k}_3 \\ \tilde{k}_2 \end{bmatrix} \right) \in R^{m \times 2m} \]

\[ D_c = \text{diag} \left( \begin{bmatrix} \tilde{k}_3^m \\ \tilde{k}_2^m \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{k}_3 \\ \tilde{k}_2 \end{bmatrix} \right) \in R^{m \times m} \]

With the multi-loop PID controller (11) the closed-loop system can be written as (assuming \( r = 0 \)):

\[ X_{k+1} = (A_0 + B_0 K C_0) X_k + (C_0 + B_0 K F_0) \zeta_k \]

\[ z_k^s = (\bar{C}_i + \bar{D}_i K C_0) X_k + \bar{D}_i K F_0 \zeta_i, \quad i = 1 \ldots l \]  

(13)
where $X_k = \begin{bmatrix} x_k \\ x_k^m \end{bmatrix}$ and the matrices $A_0, B_0, C_0, G_0, F_0, K, \bar{C}_i$ and $\bar{D}_i$ are composed as follows:

$$
\begin{align*}
A_0 &= \begin{bmatrix} A & 0 \\ -B_cC & A_c \end{bmatrix}, \\
B_0 &= \begin{bmatrix} B \\ 0 \end{bmatrix}, \\
C_0 &= \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}, \\
G_0 &= \begin{bmatrix} G \\ -B_cF \end{bmatrix}
\end{align*}
$$

(14)

To make the closed-loop system (13) satisfy the generalized covariance constraints we use the following lemma from [10]:

**Lemma 3.1** The closed-loop system (13) is stable and satisfies constraints (4) if and only if there exists a matrix $\Sigma > 0$ such that

$$
\begin{align*}
(A_0 + B_0KC_0) \Sigma (A_0 + B_0KC_0)^T - \Sigma + (G_0 + B_0KF_0) \Omega (G_0 + B_0KF_0)^T < 0 \\
(C_i + \bar{D}_iKC_0) \Sigma (C_i + \bar{D}_iKC_0)^T + \bar{D}_iKF_0\Omega F_0^T K^T \bar{D}_i^T < \Phi_i
\end{align*}
$$

(15) (16)

where $i = 1, ..., l$.

### 4 Computational algorithm

Inequalities (15) and (16) are difficult to solve and the difficulty lies in two facts: first, both Inequality (15) and Inequality (16) contain cubic terms; second, the decentralized control structure of the multi-loop PID controller makes the unknown matrix $K$ a sparse matrix. To solve the nonlinear matrix inequalities (15) and (16), one need change them to some equivalent forms that can be solved. By applying the Schur complement Lemma, it can be obtained:

**Proposition 4.1** The discrete-time system (1) is stabilized by a multi-loop PID controller, defined in (11), and the constraints (4) are satisfied if and only if there exist matrices $X > 0, Y > 0$ and $K$ (the decision variable $K$ is composed as that in (14)) such that

$$
\begin{align*}
&\begin{bmatrix}
-X & A_0 + B_0KC_0 & G_0 + B_0KF_0 \\
(A_0 + B_0KC_0)^T & -Y & 0 \\
(G_0 + B_0KF_0)^T & 0 & -\Omega^{-1}
\end{bmatrix} < 0 \\
&(\bar{C}_i + \bar{D}_iKC_0) \Omega \begin{bmatrix} \bar{C}_i & \bar{D}_iKC_0 & \bar{D}_iKF_0 \\
\bar{D}_i^T & 0 & \Omega^{-1}
\end{bmatrix} > 0 \\
&XY = I_{n+2m}
\end{align*}
$$

(17) (18) (19)

where $i = 1, ..., l$.

The proof is straightforward and it is omitted here. Obviously the condition in the proposition (4.1) is not convex because $X > 0$ and $Y > 0$ are inverse to each other. To find a feasible
solution to (17) - (19) the idea of the cone complementary linearization method [6] is adopted. The algebraic equation (19) is relaxed with the following LMI:

\[
\begin{bmatrix}
X & I_{n+2m} \\
I_{n+2m} & Y
\end{bmatrix} \geq 0
\]

and the linearized version of \( \text{trace}(XY) \) is minimized at each step.

The algorithm to calculate the multi-loop PID controller is summarized as follows:

**Algorithm 4.1**

1. Set \( k=0 \). Initialize \( X_k > 0 \in \mathbb{R}^{n+2m} \), \( Y_k > 0 \in \mathbb{R}^{n+2m} \)

2. Find \( X_{k+1} > 0 \in \mathbb{R}^{n+2m} \), \( Y_{k+1} > 0 \in \mathbb{R}^{n+2m} \) that solve the following semi-definite programming problem:

   \[
   \minimize_{X,Y,K} \text{trace}(X_k Y + Y_k X) \quad \text{subject to} \ (17), (18), (20)
   \]

3. Set \( t_k = \text{trace}(X_k Y_{k+1} + Y_k X_{k+1}) \).

4. Set \( k = k + 1 \). If the decrease of \( t_k \) in last \( L \) steps is less than a small constant number \( \varepsilon_1 > 0 \), then the algorithm stops. If \( \text{trace}(X_k Y_k) - n - 2m < \varepsilon_2 \) then go to step 4; otherwise, go to step 2.

4. Find \( \Sigma > 0 \) by solving LMI (15) and LMI (16) (where \( K \) is obtained from step 3). If there is a solution, then one feasible solution is found; otherwise, go to step 2.

**Remark 4.1** The above computational algorithm is an extension of the one so called cone complementarity linearization algorithm presented in [6] by introducing the sparse matrix \( K \) into inequalities (17) and (18) as decision variable. Similar to the proof of Theorem (2.1) in [6] it can be shown that \( t_k \) decreases with each step so that the algorithm converges.

**Claim 4.1** The Algorithm (4.1) is convergent.

5 Process with random walk disturbance

In process industry random walk disturbance is often used to represent slow dynamic of disturbances. The algorithm presented in the last section can not deal with random walk disturbance. This is because the closed-loop system, composed by the multi-loop PID controller and the process, is not a minimal realization. To generate a minimal realization of the closed-loop system, one needs to change the process diagram, and the procedure is illustrated by using a univariate feedback control example. A block diagram of a single-loop feedback system is shown in Figure (2):
where \( G(z) \) is the process model and \( \frac{H(z)}{z-1} \) is the disturbance model. Since the PID controller block and this disturbance block both contain a pole at 1, this unstable pole is moved out of these two blocks. The reconfigured block diagram is shown in Figure (3).

\[
\begin{align*}
C(z) & = k_1 + k_2 + k_3 \zeta^2 + (-k_1 - 2k_2)z + k_3 \\
G_1(z) & = G(z)z \\
H(z) & = \frac{1}{z-1}
\end{align*}
\]

The PID controller “borrows” a pole (located at the origin) from the process in order to preserve its properness. The zero order hold ensures that the process model \( G(z) \) is strictly proper, in other words, \( G(z) \) can always have an extra pole that is “lent” to the controller block. After this block diagram reconfiguration, a state space representation of the controller \( C(z) \) is:

\[
\begin{align*}
x_{k+1} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e_k \\
u_k &= \begin{bmatrix} k_3 & -k_1 - 2k_2 \end{bmatrix} x_k + (k_1 + k_2 + k_3) e_k
\end{align*}
\]  

Following the same procedure as in Section 3, we can build the controller and the closed-loop system in state space for the MIMO case, and then use the algorithm (4.1) to calculate the multi-loop PID controller parameters.

**Remark 5.1** Changing the block diagram preserves the transfer function from the disturbance to the output so that only the variances for the output can be specified. The variance for the manipulated variables can not be specified because the PID controller output is not a stationary signal.
6 Simulation results

6.1 Example 1

The first example [20] is to design a single-loop PID controller for a first order plus time-delay process subject to unstable disturbance containing an integrator. The process is as follows:

\[
y_k = \frac{z^{-6}}{1-0.8z^{-1}} u_k + \frac{1+0.6z^{-1}}{(1-0.6z^{-1})(1+0.7z^{-1})(1-0.5z^{-1})} \Delta \zeta_k
\]  

(23)

where the series \(\{\zeta_k\}\) is white noise with zero mean and unit variance. The known minimal output variance achieved by a PID controller is 123.54 [20]. However, our algorithm shows that the variance of the output can be further reduced by optimizing the PID parameters. Using the algorithm presented in the last section, we obtain the optimal PID controller as:

\[
C(z^{-1}) = \frac{0.8021 - 1.3402z^{-1} + 0.5788z^{-2}}{1-z^{-1}}
\]  

(24)

The corresponding output variance is 87.85. Compared with the known output variance 123.54, the variance of the output is reduced by 30% by using the PID controller (24).

If let disturbance \(\zeta_k = 0\), the process (23) is actually obtained by sampling a first order plus time-delay system with sampling period 1:

\[
G(s) = \frac{5}{4.48s + 1} e^{-5s}
\]  

(25)

PID tuning for the first order plus time-delay systems is well studied and there are many tuning rules available. The PID controllers, obtained using these tuning algorithms, and the corresponding output variance are listed in Table 1. The PID controllers are calculated according to the Tables 15.2, 15.3, 15.4 and 15.6 in [19].

<table>
<thead>
<tr>
<th>Tuning methods</th>
<th>PID controller</th>
<th>output variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ziegler-Nichols method</td>
<td>(0.7742-1.2903z^{-1}+0.5376z^{-2})</td>
<td>105.87</td>
</tr>
<tr>
<td>Cohen-Coon</td>
<td>(0.759-1.163z^{-1}+0.437z^{-2})</td>
<td>112.22</td>
</tr>
<tr>
<td>IMC ((\lambda = 0.2))</td>
<td>(0.6393-0.9794z^{-1}+0.3734z^{-2})</td>
<td>93.93</td>
</tr>
<tr>
<td>IMC ((\lambda = 0.4))</td>
<td>(0.548-0.839z^{-1}+0.320z^{-2})</td>
<td>98.29</td>
</tr>
<tr>
<td>IMC ((\lambda = 0.6))</td>
<td>(0.478-0.735z^{-1}+0.280z^{-2})</td>
<td>107.31</td>
</tr>
<tr>
<td>Integral Time-weighted Square Error (ITAE)</td>
<td>(0.753-1.176z^{-1}+0.466z^{-2})</td>
<td>93.63</td>
</tr>
<tr>
<td>Our algorithm</td>
<td>(0.8021-1.3402z^{-1}+0.5788z^{-2})</td>
<td>87.85</td>
</tr>
</tbody>
</table>

Table 1: PID controllers and the corresponding output variances

The PID design algorithm in this paper is proposed for MIMO systems, but this example shows that it can be used for SISO systems. The obtained PID controller (24) has the best performance among all the PID controllers.
6.2 Example 2

The second example is from [11]. The process is described as follows:

\[
y_k + A_1 y_{k-1} = B_0 u_{k-2} + \zeta_k
\]  
(26)

where \( A_1 = \begin{bmatrix} -0.99101 & 8.80512 \cdot 10^{-3} \\ -0.80610 & -0.77089 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.89889 & -0.409329 \\ -0.56 & 0.88052 \end{bmatrix} \)

and the series \( \{\zeta_k\} \) is assumed to be zero mean white noise and its covariance matrix: \( \mathbb{E}(\zeta_i \zeta_j^T) = \delta(i-j)0.01 \times \mathbb{I}_2 \).

The process is controlled by the conventional GMVC [11]:

\[
u_k = -\begin{bmatrix} 1.8989 + 1.9957z^{-1} & -0.4093 - 0.4134z^{-1} \\ -0.56 + 0.2929z^{-1} & 1.8805 + 1.4488z^{-1} \end{bmatrix} \begin{bmatrix} 0.975 & -0.0155 \\ 1.4203 & 0.5872 \end{bmatrix} y_k
\]  
(27)

With the controller (27) implemented on the process, the covariance for the output \( y_k \) is \( \begin{bmatrix} 0.0621 & 0.0782 \\ 0.0782 & 0.1411 \end{bmatrix} \) and the covariance for the input \( u_k \) is \( \begin{bmatrix} 0.0169 & 0.0255 \\ 0.0255 & 0.0413 \end{bmatrix} \).

To use the algorithm in Section 4, the state space model is first generated from (26):

\[
x_{k+1} = \begin{bmatrix} -A_1 & B_0 \\ 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ I_2 \end{bmatrix} u_k + \begin{bmatrix} -A_1 \\ 0 \end{bmatrix} \zeta_k
\]

\[
y_k = \begin{bmatrix} I_2 & 0 \end{bmatrix} x_k + \zeta_k
\]  
(28)

where \( x_k = \begin{bmatrix} y_k - \zeta_k \\ u_{k-1} \end{bmatrix} \).

The obtained multi-loop PID control is as follows:

\[
C\left(z^{-1}\right) = \begin{bmatrix} 0.5647 - 0.5387z^{-1} + 0.02z^{-2} \\ 0.3815 - 0.5476z^{-1} + 0.1678z^{-2} \end{bmatrix} \]

\[
\frac{z^{-1}(1-z^{-1})}{z^2(1-z^{-1})}
\]  
(29)

With the multi-loop PID controller (29) implemented on the process, the covariance for the simulated \( y_k \) is \( \begin{bmatrix} 0.0249 & 0.0224 \\ 0.0224 & 0.1158 \end{bmatrix} \) and the covariance for the simulated \( u_k \) is \( \begin{bmatrix} 0.0084 & 0.0041 \\ 0.0041 & 0.0075 \end{bmatrix} \).

It can be seen that by using the multi-loop PID controller the variances of the process input and output are smaller than those by using the GMVC. The performance comparison of multi-loop PID and GMVC is shown in Figure (4).

6.3 Example 3

The third example is a dry rotary cement kiln with capacity 1000 tons/day [15]. The kiln is 105 meters long and 5 meters in diameter. After two pre-heaters, where the dry homogenized raw material is heated to 800°C, then goes to the kiln. The dry homogenized raw material enters the kiln and then passes it. The final temperature of the material is around 1450°C. Coal is burned in the lower front end of the kiln in order to produce the high temperature, which is required to start the chemical reactions taking place in the raw materials. The product of the
reaction is called clinker that is cooled in a planetary cooler before it leaves the process. The kiln process has two controlled variables: the combustion gas temperature and the kiln drive power. The latter is chosen as a controlled variable because it correlates to the burning temperature and clinker quality and the clinker quality can only be analyzed every two hours. The two manipulated variables of the kiln process are the kiln exhaust fan speed and raw material feed rate. The process is exposed to random disturbance. The sampling period is 5 minutes. The original process model is shown as follows:

\[ y_{k+1} + A_0 y_k = B_0 u_k + \zeta_{k+1} + C_0 \zeta_k \]  

(30)

where

\begin{align*}
A_0 &= \begin{bmatrix} -0.917 & -0.0846 \\ 0.132 & -0.915 \end{bmatrix}, & B_0 &= \begin{bmatrix} 2.06 \\ -0.108 \end{bmatrix} \\
C_0 &= \begin{bmatrix} -0.0449 & -0.216 \\ 0.0256 & 0.841 \end{bmatrix}, & E(\zeta_i \zeta_j^T) &= \begin{bmatrix} 0.0639 & 0.00188 \\ 0.00188 & 0.0233 \end{bmatrix} \delta(i-j) 
\end{align*}

One can obtain the following state space model from the kiln process (30):

\begin{align*}
x_{k+1} &= \begin{bmatrix} 0.917 & 0.0846 \\ -0.132 & 0.915 \end{bmatrix} x_k + \begin{bmatrix} 2.06 \\ -0.108 \end{bmatrix} u_k + \begin{bmatrix} 0.8721 \\ -0.1064 \end{bmatrix} \zeta_k \\
y_k &= x_k + \zeta_k
\end{align*}

(31)

It is desired to control the variances of both the output and input so that the controlled variables
are chosen as:

\[
\begin{align*}
z_k^1 &= \begin{bmatrix} 1 & 0 \end{bmatrix} y_k \\
z_k^2 &= \begin{bmatrix} 0 & 1 \end{bmatrix} y_k \\
z_k^3 &= \begin{bmatrix} 1 & 0 \end{bmatrix} u_k \\
z_k^4 &= \begin{bmatrix} 0 & 1 \end{bmatrix} u_k
\end{align*}
\] (32)

It is shown in [15] that a reasonable control criterion is to minimize the joint variation of the controlled variables:

\[
J = \lim_{k \to \infty} E y_k^T y_k
\] (33)

However, the variances of the input variables become unacceptable if the minimum variance control law is implemented: \(\lim_{k \to \infty} E (z_k^3 z_k^3) = 0.148\) and \(\lim_{k \to \infty} E (z_k^4 z_k^4) = 108\). As it is pointed out in [15] it is appropriate to restrict the variances of the input variables according to:

\[
\begin{align*}
\lim_{k \to \infty} E (z_k^3 z_k^3) &< 0.004 \\
\lim_{k \to \infty} E (z_k^4 z_k^4) &< 1.5
\end{align*}
\] (34)

Minimization of the (33) subject to the variance constraints (34) by using a full order dynamic controller leads to the output variances [10, 15]:

\[
\begin{align*}
\lim_{k \to \infty} E (z_k^1 z_k^1) &= 0.0939 \\
\lim_{k \to \infty} E (z_k^2 z_k^2) &= 0.189
\end{align*}
\] (35)

The proposed algorithm can not find a multi-loop PID controller if the variance constraints for the input variables are chosen as (34) and the variance constraints for the output variables are specified as 0.939 and 0.189 respectively. This is not surprising because the decentralized controller structure adds performance limit compared to the full order centralized controller. If the variance constraint for \(z_k^2\) is relaxed to 0.345, by using the algorithm in Section 4 a multi-loop PID controller can be obtained as follows:

\[
C \left( z^{-1} \right) = \begin{bmatrix} 0.1743-0.1612z^{-1}+0.0064z^{-2} \\ -(1-z^{-1})z^{-1} \end{bmatrix}
\] (36)

The simulation results are shown in Figure (5):

With the multi-loop PID controller (36) implemented, the variances for the controlled variables, calculated from the simulated data, are:

\[
\begin{align*}
\lim_{k \to \infty} E z_k^1 z_k^1 &= 0.0923 \\
\lim_{k \to \infty} E z_k^3 z_k^3 &= 0.3419
\end{align*}
\]

It is shown that the variance for the first output is even slightly better than that achieved by the constrained LQG controller in [15]; the variances of the input variables are better than those achieved by the constrained LQG controller. However, the variance of \(z_k^2\) is larger than that achieved by the constrained LQG controller in [15] but still satisfies the design specification. Simulation also shows that if the variance bound for the second output is smaller than 0.33, then
the algorithm can not find a solution. This may be caused by two reasons: 1) the multi-loop-PID-controller structure is far simpler than that of the full-order multivariable controller, and this simplicity will reduce the closed-loop performance compared to the full-order optimal constrained LQG controller in [15]; 2) the proposed algorithm may not be globally convergent, which means that if there exists a multi-loop PID controller such that the variance $z_k^2$ is smaller than 0.33, the proposed algorithm may not find it. There is no strict proof of the global convergency of the proposed algorithm in Section 4; however, our simulation shows the algorithm always converges to one value no matter what is the initial condition. It is worth a investigation of the global convergent properties of the proposed algorithm as a future research topic.

7 Conclusion

The multi-loop or decentralized PID controller design based on the generalized covariance constraints has been considered in this paper and an iterative LMI approach is proposed to solve the problem. The algorithm is shown to be convergent. This algorithm is originally derived for the process with stable disturbances; after the reconfiguration of the process block diagram, it can also be applied to the process with unstable random-walk disturbances. Several simulation results are used to illustrate the effectiveness of the proposed method.

References


