Family of the PID Controllers

Ilan Rusnak
RAFAEL, Advanced Defense Systems, Haifa
Israel

1. Introduction

The PID controllers (P, PD, PI, PID) are very widely used, very well and successfully applied controllers to many applications, for many years, almost from the beginning of controls applications (D’Azzo & Houpis, 1988)(Franklin et al., 1994). (The facts of their successful application, good performance, easiness of tuning are speaking for themselves and are sufficient rational for their use, although their structure is justified by heuristics: "These ... controls - called proportional-integral-derivative (PID) control - constitute the heuristic approach to controller design that has found wide acceptance in the process industries." (Franklin et al., 1994, pp. 168)).

In this chapter we state a problem whose solution leads to the PID controller architecture and structure, thus avoiding heuristics, giving a systematic approach for explanation of the excellent performance of the PID controllers and gives insight why there are cases the PID controllers do not work well. Namely, by the use of Linear Quadratic Tracking (LQT) theory (Kwakernaak & Sivan, 1972)(Anderson & Moore, 1989) control-tracking problems are formulated and those cases when their solution gives the PID controllers are shown.

Further, problem of controlling-tracking high order polynomial inputs and rejecting high order polynomial disturbances is formulated. By applying the LQT theory extended family of PID controllers – the family of generalized PID controllers denoted $\text{PI}^m_D^{n-1}$ is derived. This provides tool for application of optimal controllers for those systems that the conventional PID controllers are not satisfactory, for generalization and derivation of further results. The notation of generalized PID controllers, $\text{PI}^m_D^{n-1}$, is consistent with the notation of controllers for fractional order systems (Podlubny, 1999).

The present work is strongly motivated by problems-question tackled by the author during a continuous work on high performance servo and motion control applications. Some of the theoretical results that have had motivated and led to the present work have been documented in (Rusnak, 1998, 1999, 2000a, 2000b). Some of the presented architectures appear and are recommended for use in (Leonhard, 1996, pp. 80, 347) without rigorous rationale and were partial trigger for the presented approach.

By Architecture we mean, loosely, the connections between the outputs/sensors and the inputs/actuators; Structure deals with the specific realization of the controllers' blocks; and Configuration is a specific combination of architecture and structure. These issues fall within the control and feedback organization theory that have been reviewed and presented in a concise form in (Rusnak, 2002, 2005) and in a widened form in (Rusnak, 2006, 2008). It is beyond the scope of this chapter. It is used here as a basis at a system theoretic level to
enable formulation of the control-feedback loops organization problem that leads to the family of generalized PID controllers. This article does not deal with the numerical values of the controllers' filters coefficients/gains; rather it concentrates in organization of the control loop and structure of the filters. This is the way the optimal LQT theory is used.

The LQT theory requires a reference trajectory generator. The reference trajectory is generated by a system that reflects the nominal behavior of the plant. The differences are the initial conditions, the input to the reference trajectory generator and the deviation of the actual plant from the nominal one. The zero steady state error is imposed by integral action of a required order on the state tracking error.

The generalized controllers derived by the presented methodology have been applied to high performance motion control in (Nanomotion, 2009a, 2009b) and to high performance missile autopilot in (Rusnak and Weiss, 2011).

The novelty of the results in this approach is that it shows for what problems a controller from the family of PID controllers is the optimal controller and for which it is not. The importance of this result is:

1. From theoretical point of view it is important to know that widely used control architecture can be derived from an optimal control/tracking problem.
2. The solution shows for what kind of systems the PID controller is optimal and for which systems it is not, thus showing why a PID controller does not perform well for all systems. This will enable to forecast what control designs not to apply a PID controller.
3. For those systems that the PID is not the controller architecture derived from the optimal control approach shows what is the optimal controller architecture and structure, thus achieving generalization.
4. The present approach advises how to design PID controller on finite time interval, i.e. when the gains are time varying.
5. The generalization can be used in deriving generalized PID controllers for high order SISO systems, for SIMO and MIMO systems (Rusnak, 1999, 2000a), for time-varying and non-linear systems; thus enabling a systematic generalization of the PID controller paradigm.
6. The design procedures of PID controllers are assuming noise free environment. The presented approach advises how to generalize the PID controller configuration in presence of noises by the use of the Linear Quadratic Gaussian Tracking-LQGT theory (Rusnak, 2000b).
7. The conventional PID paradigm introduces integral action in order to drive the steady state tracking errors in presence of constant reference trajectory or disturbance. The present approach enables to systematically generalize the controller to drive the steady state tracking errors to zero for high order polynomial inputs and disturbances.
8. Choosing the optimal generalized PID controller reduces the quantity of controller parameters-gains that are required for tuning. Thus saving time during the design process.
9. The LQT motivated architecture enables separate treatment of the transient, by the trajectory generator, and the steady state performance by introducing integrators into the controllers (Rusnak and Weiss, 2011).

The results on the architecture and structure of the PID controllers' family for 1st and 2nd order are rederived in the article. Specifically, it is shown that the classical one block PID controller is optimal for Linear Quadratic Tracking problem of a 2nd order minimum phase plant. For plants with non-minimum phase zero the family of PID controllers is only suboptimal. Multi output single input architectures are proposed that are optimal.
Throughout this chapter the same notation for time domain and Laplace domain is used, and the explicit Laplace variable (s) is stated to avoid confusion wherever necessary.

2. The optimal tracking problem

The optimal tracking problem is introduced in (Kwakernaak & Sivan, 1972) (Anderson & Moore, 1989). The n\textsuperscript{th} order system is

\[
\dot{x} = Ax + Bu; \quad x(t_o) = x_o, \\
y = Cx
\]

where \(x\) is the state; \(u\) is the input and \(y\) is the measured output, \(x_o\) is a zero mean random vector.

The \(v\)\textsuperscript{th} order reference trajectory generator is

\[
\dot{x}_r = A_r x_r + B_r w_r; \quad x_r(t_o) = x_{ro}, \\
y_r = C_r x_r
\]

where \(x_r\) is the state; \(w_r\) is the input and \(y_r\) is the reference output; \(w_r\) is a zero mean stochastic process, \(x_{ro}\) is zero mean random vector. Further it is assumed that \(n=v\). The case \(n\neq v\) is beyond the scope of this chapter.

The integral action is introduced into the control in order to “force” zero tracking errors for polynomial inputs, and to attenuate disturbances (Kwakernaak & Sivan, 1972)(Anderson & Moore, 1989). This is done by introducing the auxiliary variables, integrals of the tracking error. This way the generalized PID controller, denoted \(PI^{n-1}_m\), is created. That is, the state is augmented by

\[
\begin{align*}
    e_s &= x_r - x \\
    \dot{\eta}_1 &= C_{e1} [x_r - x] = C_{e1} e_x \\
    \dot{\eta}_2 &= C_{e2} \dot{\eta}_1; \\
    \vdots \\
    \dot{\eta}_m &= C_{em} \eta_{m-1} \\
    \eta &= \begin{bmatrix} \eta_1 \\
                     \eta_2 \\
                     \vdots \\
                     \eta_m \end{bmatrix} \\
    \eta(t_o) &= \eta_o,
\end{align*}
\]

where \(m\) is the number of integrators that are introduced on the tracking error.

The control objective is

\[
J = \frac{1}{2} \mathbb{E} \left\{ \int_{t_o}^{t_f} \left[ \begin{array}{c}
    y_r(t_f) - y(t_f) \\
    \eta(t_f) \end{array} \right]^T G_1 \left[ \begin{array}{c}
    y_r(t_f) - y(t_f) \\
    \eta(t_f) \end{array} \right] + \eta(t_f)^T G_2 \eta(t_f) \\
    + \int_{t_o}^{t_f} \left[ \begin{array}{c}
    y_r(t) - y(t) \\
    \eta(t) \end{array} \right]^T Q_1 \left[ \begin{array}{c}
    y_r(t) - y(t) \\
    \eta(t) \end{array} \right] + \eta(t)^T Q_2 \eta(t) + u(t)^T R u(t) \right\}
\]

www.intechopen.com
The optimal tracking problem (Kwakernaak & Sivan, 1972) is to find an admissible input $u(t)$ such that the tracking objective (5) is minimized subject to the dynamic constraints (1-4).

All vectors and matrices are of the proper dimension.

3. Solution of the optimal tracking problem

In order to solve the Optimal Tracking Problem we augment the state variables (Kwakernaak & Sivan, 1972) and further assume that $A=\bar{A}$, $B=\bar{B}$, and $C=\bar{C}$. This assumption states that the nominal values of the plant's parameters are known. The case $A\neq\bar{A}$, $B\neq\bar{B}$, and $C\neq\bar{C}$, is beyond the scope of this chapter.

We have the error system

$$\dot{e}_x = A e_x + B w_r - B e; \quad e_x(t_o) = x_o - x_r,$$

$$X = \begin{bmatrix} e_x \\ n_1 \\ n_2 \\ \vdots \\ n_m \end{bmatrix}; \quad \bar{A} = \begin{bmatrix} A & 0 & 0 & \cdots & 0 \\ C_{e_1} & 0 & 0 & \cdots & 0 \\ 0 & C_{e_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{en_m} & 0 \end{bmatrix}; \quad \bar{B} = \begin{bmatrix} -B \\ 0 \end{bmatrix}; \quad \bar{C} = [C \ 0 \ 0 \ 0]$$

then the problem is minimization of (5) subject to (1-4) is the problem of minimization of

$$J = \frac{1}{2} E [X(t_f)^T G X(t_f) + \int_{t_o}^{t_f} [X(t)^T Q X(t) + u(t)^T R u(t)] dt]$$

subject to

$$X = \bar{A} X + \bar{B} u - \bar{B} \bar{w}_r, \quad \bar{x}(t_o) = \bar{x}_o,$$

where

$$Q = \bar{C}^T Q \bar{C} + \begin{bmatrix} 0 & 1 \end{bmatrix} Q_2 \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad G = \bar{C}^T G \bar{C} + \begin{bmatrix} 0 & 1 \end{bmatrix} G_2 \begin{bmatrix} 0 & 1 \end{bmatrix}.\quad (10)$$

The solution is (Kwakernaak & Sivan, 1972) (Bryson & Ho, 1969)

$$u = -R^{-1} \bar{B} P X$$

$$-\dot{P} = P \bar{A} + \bar{A}^T P + Q - P \bar{B} R^{-1} \bar{B} P, \quad P(t_f) = G.$$\quad (11)

If we appropriately partition $P$, then

$$u = R^{-1} B \begin{bmatrix} P_{11} & P_{12} \end{bmatrix} \begin{bmatrix} e_x \\ \eta \end{bmatrix} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} e_x \\ \eta \end{bmatrix}$$\quad (12)

Notice that the solution is independent of the reference trajectory generator input, $\bar{w}_r$. 

www.intechopen.com
4. Architectures

As stated in the introduction Architecture deals with the connections between the outputs/sensors and the inputs/actuators; Structure deals with the specific realization of the controllers' blocks; and Configuration is a specific combination of architecture and structure. These issues fall within the of control and feedback organization theory (Rusnak, 2006, 2008), and are beyond the scope of this chapter.

In this chapter we deal with three specific architectures. These are:

1. Parallel controller architecture;
2. Cascade controller architecture;
3. One block controller architecture.

4.1 Parallel controller architecture

This control architecture is directly derived from the Solution of the Optimal Tracking Problem as derived in (Asseo, 1970) and in (12). The parallel controller can be written directly from (12) in Laplace domain as

\[ u(s) = \sum_{i=1}^{n} C_i(s) (x_{r_i}(s) - x_i(s)) = \sum_{i=1}^{n} C_i(s) e_i(s) \]

(13)

For 2\textsuperscript{nd} order system the parallel controller architecture takes the form.

\[ u(s) = C_1(s)(x_{r_1}(s) - x_1(s)) + \ldots + C_n(s)(x_{r_n}(s) - x_n(s)) \]

(14)

Figure 1 presents the block diagram of the parallel controller architecture for a 2\textsuperscript{nd} order system.

![Parallel controller architecture for 2\textsuperscript{nd} order system.](image)

4.2 Cascade controller architecture

By elementary block operation (13) can be written as

\[ u(s) = C_n(s)[(x_{r_n}(s) - x_n(s)) + \frac{C_{n-1}(s)}{C_n(s)}[(x_{r_{n-1}}(s) - x_{n-1}(s)) + \ldots + \frac{C_{n-2}(s)}{C_{n-1}(s)}[(x_{r_{n-2}}(s) - x_{n-2}(s)) + \ldots \ldots \ldots \ldots (15)\]

\[ + \frac{C_2(s)}{C_1(s)}(x_2(s) - x_1(s)) ... ] \]

For 2\textsuperscript{nd} order system: \[ u(s) = C_2(s)(x_{r_2}(s) - x_2(s)) + \ldots \]

(14)
This is the cascade controller architecture. For 2\textsuperscript{nd} order system the cascade controller architecture takes the form.

\[ u(s) = C_2 \left\{ (x_{r2} - x_2) + \frac{C_1}{C_2} (x_{r1} - x_1) \right\} = C_v \{(x_{r2} - x_2) + C_p (x_{r1} - x_1)\} \]  

(16)

Figure 2 presents the block diagram of the cascade controller architecture for a 2\textsuperscript{nd} order system. The rationale for the notation of \( C_p \) (position) and \( C_v \) (velocity) will be presented in the sequel.

4.3 One block controller architecture

By elementary operation on (13), and exploiting the relations between the state space variables, the one block controller architecture can be written as

\[ u(s) = C(s) \left\{ x_{r1}(s) - x_1(s) \right\} \]  

(17)

Figure 3 presents the block diagram of the one block controller architecture.

4.4 Discussion

Although from input-output transfer function point-of-view, there is no formal difference between the different architectures, there is difference with respect to the response to initial conditions, effects of saturation and nonlinearities, robustness, and more.
5. Controllers for first order system

As a first order system is considered, this leads to the one block controller architecture only.

5.1 P controller

Here we have

\[
u = k_1 e_x = k_1 (x_r - x) \tag{18}\]

This is the proportional - P controller.

\[C(s) = k_1\]

5.2 PI controller

Here we have

\[
\dot{\eta}_1 = e_x \\
u = R^{-1}B^T \begin{bmatrix} P_{11} & P_{12} \end{bmatrix} \begin{bmatrix} e_x & \eta \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} e_x \\ \eta_1 \end{bmatrix} = k_1 e_x + k_2 \int e_x dt \tag{20}\]

This is the proportional + Integral - PI controller.

\[C(s) = k_1 + \frac{k_2}{s} = \frac{k_1 s + k_2}{s} \tag{21}\]

5.3 PI² controller

Here we have

\[
\dot{\eta}_1 = e_x \\
\dot{\eta}_2 = \dot{\eta}_1, \text{ or } \dot{\eta}_2 = e_x \\
u = R^{-1}B^T \begin{bmatrix} P_{11} & P_{12} \end{bmatrix} \begin{bmatrix} e_x & \eta \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_{21} \\ k_2 & k_{22} \end{bmatrix} \begin{bmatrix} e_x \\ \eta_1 \\ \eta_2 \end{bmatrix} = k_1 e_x + k_{21} \int e_x dt + k_{22} \int \int e_x dt \tag{22}\]

This is the proportional + double integrator - PI² controller.

\[C(s) = k_1 + \frac{k_{21}}{s} + \frac{k_{22}}{s^2} = \frac{k_1 s^2 + k_{21} s + k_{22}}{s^2} \tag{23}\]

5.4 PIⁿ controller

Here we have

\[
\dot{\eta}_1 = e_x \\
\dot{\eta}_2 = \eta_1, \text{ or } \dot{\eta}_2 = e_x, \\
\vdots \\
\dot{\eta}_m = \eta_{m-1}, \text{ or } \dot{\eta}_m = e_x \\
\dot{\eta}_{m+1} = \eta_m, \text{ or } \dot{\eta}_{m+1} = e_x \tag{24}\]
\[
    u = R^{-1}B^T \begin{bmatrix} P_{11} & P_{12} \end{bmatrix} \begin{bmatrix} e_x \ 
    \eta \end{bmatrix} = \begin{bmatrix} k_1 & k_{21} & k_{22} & \ldots & k_{2m} \end{bmatrix} \begin{bmatrix} e_x \\ \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix}
\] (25)

\[
    = k_i e_x + k_{21} \int e_x \, dt + \ldots + k_{2m} \int e_x \, dt \int e_x \, dt \int e_x \, dt 
\]

This is the proportional + (m) integrators - PI^m controller.

\[
    C(s) = k_1 + \frac{k_{21}s + k_{22}}{s} + \ldots + \frac{k_{2m} s^m + k_{21} s^{m-1} + \ldots + k_{2m}}{s^m} 
\] (26)

Table 1 summarizes the one block generalized PID controller structure for first order system.

<table>
<thead>
<tr>
<th>controller</th>
<th>expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>( k_1 )</td>
</tr>
<tr>
<td>PI</td>
<td>( \frac{k_1s + k_2}{s} )</td>
</tr>
<tr>
<td>PI^2</td>
<td>( \frac{k_1s^2 + k_{21}s + k_{22}}{s^2} )</td>
</tr>
<tr>
<td>PI^m</td>
<td>( \frac{k_1s^m + k_{21}s^{m-1} + \ldots + k_{2m}}{s^m} )</td>
</tr>
</tbody>
</table>

Table 1. One block generalized PID controllers for 1st order system.

### 6. Controllers for second order system

Second order plant and the trajectory generator are assumed and are represented in the companion form

\[
    A = A_r = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad C = C_r = [1 \ 0], 
\] (27)

and we have

\[
    H(s) = H_r(s) = \frac{x_1}{u} = \frac{y}{u} = \frac{b_1s + (b_2 + a_1b_1)}{s^2 + a_1s + a_2}, 
\] (28)

\[
    \frac{x_2}{u} = \frac{b_2s - a_2b_1}{s^2 + a_1s + a_2} 
\] (29)

\[
    \frac{x_2}{x_1} = \frac{b_2s - a_2b}{b_1s + (b_2 + a_1b_1)} 
\] (30)

The plant's and trajectory's state generator are denoted...
The reason for selecting the state space representation (27) is that plant without zero, i.e. $b_1 = 0$, is a case that is often met in motion control with electrical and PZT motors (Rusnak, 2000a). For plant without zero $x_2 = \dot{y}$, so that

$$
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix},
\begin{bmatrix} x_{r1} \\
x_{r2}
\end{bmatrix} = \begin{bmatrix} y_r \\ \dot{y}_r \end{bmatrix}
$$

and one can deal with position feedback, feedback on $y$, and velocity feedback, feedback on $\dot{y}$. For this reason in this chapter we will call, with slight abuse of nomenclature, the feedback loop on $y$ the position loop and the feedback loop on $x_2, (\hat{y})$, the velocity loop.

### 6.1 PD controller
Feedback without integral action is implemented. The tracking errors are

$$
e_{x1} = y_r - y = x_{r1} - x_1 = e
$$

$$
e_{x2} = x_{2r} - x_2
$$

The controller is

$$
u = [k_1 \quad k_2] \begin{bmatrix} e_{x1} \\ e_{x2} \end{bmatrix} = k_1 e + k_2 (x_{2r} - x_2)
$$

#### 6.1.1 Parallel controller

$$
u = k_1 (y_r - y) + k_2 (x_{2r} - x_2)
$$

#### 6.1.2 Cascade controller

$$
u = k_2 \left( x_{2r} - x_2 \right) + \frac{k_1}{k_2} (y_r - y)
$$

#### 6.1.3 One block controller
To get the one block controller we substitute (30) and get (in Laplace domain)

$$
u(s) = k_1 e + k_2 (x_{2r} - x_2) = k_1 e + \frac{k_2 x_2}{s} \left( x_{1r} - x_1 \right)
$$

$$
C(s) = \frac{u(s)}{e(s)} = k_1 + k_2 \frac{b_2 s - a_2 b}{b_1 s + (b_2 + a_1 b_1)} = k_p + \frac{k_D s}{s T_D + 1}
$$
This is the PD controller.

6.1.4 Discussion
1. We used the assumption that \( x_2(s) / x_1(s) = x_{2r}(s) / x_{1r}(s) \) and ignored the response to initial conditions.
2. For 2nd order plant with a stable zero the optimal controller is a proper PD controller, i.e. no direct derivative is required.
3. The pole/filter of the derivative in (37) cancels out the zero of the plant (28). This is optimal/correct for deterministic (noiseless) systems. For systems with noisy measurements this cancelation is no more optimal (Rusnak, 2000b).
4. The cancellation of the plant’s zero by the optimal controller creates an uncontrollable system. This may work (although is not good practice) for stable zero. However, when the plant has non-minimum phase (unstable) zero the optimal PD controller induces uncontrollable unstable mode, which means that the Optimal PD controller cannot/should not be implemented in the one block controller architecture.
5. As for a plant with unstable zero the optimal one block PID controller cannot be realized, then measurement of the two states, or an observer is required if one wishes to build the optimal controller.
6. If stable controller is required it is possible to implement the optimal PD one block architecture controller only for minimum phase plants!
7. For 2nd order system without zero the deterministic optimal controller is not proper, i.e. requires pure derivative.

6.2 PID controller
Zero steady state tracking error on the output is required. The tracking errors are

\[
e_{x_1} = y_r - y_r = x_{1r} - x_1 = e \\
e_{x_2} = x_{2r} - x_2 \\
\dot{e}_{x_1} = e_{x_1}
\]

(38)

The controller is

\[
u = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} e_{x_1} \\ e_{x_2} \\ \dot{e}_{x_1} \end{bmatrix} = k_1 e + k_2 (x_{2r} - x_2) + k_3 \int e dt
\]

(39)

and the controller in Laplace domain

\[u(s) = k_1 e + k_2 (x_{2r} - x_2) + \frac{k_3}{s} e \]

(40)

6.2.1 Parallel controller

\[
u = \left( k_1 + \frac{k_3}{s} \right) (y_r - y) + k_2 (x_{2r} - x_2)
\]

(41)
6.2.2 Cascade controller

\[ u(s) = k_2 \left[ (x_{2r} - x_2) + \frac{k_4}{k_2} \left( y_r - y \right) \right] \]  

(42)

6.2.3 One block controller

To get the one block output controller derive we substitute (30) and get (in Laplace domain)

\[ u(s) = k_1 e + k_2 \left( x_{2r} - x_2 \right) + \frac{k_3}{s} e = k_1 e + k_2 \frac{x_2}{s} \left( x_{1r} - x_1 \right) + \frac{k_3}{s} e \]  

(43)

\[ C(s) = \frac{u(s)}{e(s)} = k_1 + k_2 \frac{b_2 s - a_2 b}{b_1 s + (b_2 + a_1 b_1)} + \frac{k_3}{s} = k_p + k_i + \frac{k_d}{s} \]  

This is the PID controller.

6.2.4 Discussion

1. Remarks in section 6.1.4 apply here mutatis mutandis.
2. For 2nd order plant with a stable zero, the optimal controller with one integrator is a stable proper PID controller, i.e. no direct derivative is required.

6.3 PID controller in PIV configuration

Zero steady state tracking error on the output and the second state (velocity) is required. The tracking errors are

\[ \dot{\eta}_1 = \begin{bmatrix} x_{1r} - x_1 \\ x_{2r} - x_2 \end{bmatrix}, \quad \eta_1 = \begin{bmatrix} e_{x1} dt \\ e_{x2} dt \end{bmatrix} \]  

(44)

The controller is

\[ u = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} e_{x1} \\ e_{x2} \\ \int e_{x1} dt \\ \int e_{x2} dt \end{bmatrix} = k_1 e_{x1} + k_2 e_{x2} + k_3 \int e_{x1} dt + k_4 \int e_{x2} dt \]  

(45)

and in Laplace domain

\[ u(s) = k_1 e_{x1} + k_2 e_{x2} + \frac{k_3}{s} e_{x1} + \frac{k_4}{s} e_{x2} \]  

(46)

6.3.1 Parallel controller

\[ u(s) = \left( k_1 + \frac{k_3}{s} \right) \left( y_r - y \right) + \left( k_2 + \frac{k_4}{s} \right) \left( x_{2r} - x_2 \right) \]  

(47)
6.3.2 Cascade controller - the PIV configuration

\[ u(s) = \left( k_2 + \frac{k_4}{s} \right) \left( x_{2r} - x_2 \right) + \frac{k_3 s + k_3}{k_2 s + k_4} (y_r - y) \]  (48)

This is called the PIV configuration (Proportional feedback in position loop and proportional+integral feedback in the velocity loop) (configuration=combination of architecture and structure) as there is almost proportional feedback (Lead-Lag) on the position \( x_1 \) and then in the velocity loop on \( x_2 \) there is proportional and one integral feedback.

6.3.3 One block output controller

To get the one block controller we substitute (30) and get (in Laplace domain)

\[ u(s) = \left( k_1 + \frac{k_3}{s} \right) e_{x1} + \left( k_2 + \frac{k_4}{s} \right) e_{x2} \]

\[ C(s) = \frac{u(s)}{e(s)} = \left( k_1 + \frac{k_3}{s} \right) + \left( k_2 + \frac{k_4}{s} \right) \frac{x_2}{x_1} = k_p + \frac{k_i}{s} + \frac{k_D}{s \tau_D + 1} \]  (49)

This is the PID controller.

6.3.4 Discussion

1. Remarks in section 6.1.4 apply here mutatis mutandis.
2. Two different tracking problems (38, 44) lead to the same one block controller.
3. In the parallel architecture there is a PI controller in each of the errors (47).
4. Although formally the cascade architecture controller requires the tuning of six parameters in (48), the deterministic optimal PIV controller needs the tuning of four parameters only, as can be deduced from (46).

6.4 PI²D controller

Zero steady state tracking error on the output for ramp input or disturbance is required. The tracking errors are

\[ \dot{\eta}_1 = e_{x1} = y_r - y = x_{1r} - x_1 = e \]
\[ e_{x2} = x_{2r} - x_2 \]
\[ \dot{\eta}_2 = \eta_1 \]  (50)

The controller is

\[ u = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} e_{x1} \\ e_{x2} \\ \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} e_{x1} \\ e_{x2} \\ \int e_{x1} dt \\ \int e_{x1} dt \end{bmatrix} \]

\[ = k_1 e_{x1} + k_2 e_{x2} + k_3 \int e_{x1} dt + k_4 \int \int e_{x1} dt \]  (51)

and in Laplace domain
u(s) = k_1 e + k_2 (x_{2r} - x_2) + \frac{k_3}{s} e + \frac{k_4}{s^2} e \quad (52)

6.4.1 Parallel controller

\[ u = \left( k_1 + \frac{k_3}{s} + \frac{k_4}{s^2} \right) (y_r - y) + k_2 (x_{2r} - x_2) \quad (53) \]

6.4.2 Cascade controller

\[ u(s) = k_2 \left( x_{2r} - x_2 \right) + \frac{\left( k_3 s^2 + k_3 s + k_4 \right)}{k_2 s^2} (y_r - y) \quad (54) \]

Two integrators in the position loop and proportional feedback in the velocity loop.

6.4.3 One block output controller

To get the one block controller we substitute (30) and get (in Laplace domain)

\[ u(s) = \left( k_1 + \frac{k_3}{s} + \frac{k_4}{s^2} \right) (x_{1r} - x_1) + k_2 \left( x_{2r} - x_2 \right) = \left( k_1 + \frac{k_3}{s} + \frac{k_4}{s^2} \right) (x_{1r} - x_1) + k_2 \frac{x_2}{x_1} (x_{1r} - x_1) \]

\[ C(s) = \frac{u(s)}{e(s)} = k_1 + k_2 \frac{b_2 s - a_2 b}{b_1 s + (b_2 + a_1 b)} + \frac{k_3}{s} + \frac{k_4}{s^2} = k_p + \frac{k_1}{s} + \frac{k_2}{s^2} + \frac{k_D s}{s r_D + 1} \quad (55) \]

This is the PI\textsuperscript{2}D controller.

6.4.4 Discussion

1. Remarks in section 6.1.4 apply here mutatis mutandis.

6.5 PI\textsuperscript{2}D controller in IPIV configuration

Here we want to force zero steady state tracking error on the second state, as well, however in different configuration, i.e.

\[ \eta_1 = \begin{bmatrix} x_{1r} \ x_1 \\ x_2 - x_2 \end{bmatrix}, \quad \eta_1 = \begin{bmatrix} e_x_1 \\ e_x_2 \end{bmatrix}, \quad \eta_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} e_1, \quad or \quad \eta_2 = \eta_1 = e_{x1} = e \quad (56) \]

The controller is

\[ u = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} e_{x1} \\ e_{x2} \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} e_{x1} \\ e_{x2} \end{bmatrix} \int e_{x1} dt + \int \int e_{x1} dt \quad (57) \]

\[ = k_1 e_{x1} + k_2 e_{x2} + k_3 \int e_{x1} dt + k_4 \int \int e_{x1} dt \]
and in Laplace domain

\[ u(s) = k_1 e + k_2 (x_{2r} - x_2) + \frac{k_3}{s} e + \frac{k_4}{s^2} (x_{2r} - x_2) + \frac{k_5}{s^2} e \] (58)

### 6.5.1 Parallel implementation

\[ u = \left( k_1 + \frac{k_3}{s} + \frac{k_5}{s^2} \right) (y_r - y) + \left( k_2 + \frac{k_4}{s} \right) (x_{2r} - x_2) \] (59)

### 6.5.2 Cascade controller – the IPIV configuration

\[ u(s) = \frac{k_3 s + k_4}{s} \left[ (x_{2r} - x_2) + \frac{k_1 s^2 + k_3 s + k_5}{s (k_2 s + k_4)} (y_r - y) \right] \] (60)

This is called the IPIV configuration (Proportional +integral feedback in position loop and proportional +integral feedback in the velocity loop) as there is almost proportional feedback on the position loop, \( y \), and then in the velocity loop on, \( x_{2r} \), there is proportional and one integral feedback.

### 6.5.3 One block output controller

To get the one block output controller we substitute (30) and get (in Laplace domain)

\[ u(s) = k_1 e + k_2 (x_{2r} - x_2) + \frac{k_3}{s} e + \frac{k_4}{s^2} (x_{2r} - x_2) \]

\[ C(s) = \frac{u(s)}{e(s)} = k_1 + k_2 \frac{x_2}{x_1} + \frac{k_3}{s} + \frac{k_4}{s^2} \frac{x_2}{x_1} = k_p + \frac{k_{11}}{s} + \frac{k_{12}}{s^2} + \frac{k_{2D}}{s r_D + 1} \] (61)

This is the PIFD controller.

### 6.5.4 Discussion

1. Remarks in section 6.1.4 apply here mutatis mutandis.

### 6.6 PI^2D controller in PI^2V configuration

Here we want to force zero steady state tracking error on the rate of the output as well, and

\[ \dot{x}_1 = \begin{bmatrix} x_1 - x_{1r} \\ x_2 - x_{2r} \end{bmatrix}, \quad \dot{e}_1 = \begin{bmatrix} e_{x_1} \\ e_{x_2} \end{bmatrix}, \quad \eta_1 = \begin{bmatrix} \int e_{x_1} dt \\ \int e_{x_2} dt \end{bmatrix} \]

\[ \dot{\eta}_2 = \eta_1 \] (62)

The controller is
\[ u = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 & k_6 \end{bmatrix} \begin{bmatrix} e_{x_1} \\ e_{x_2} \\ \int e_{x_1} dt \\ \int e_{x_2} dt \\ \int \int e_{x_1} dt \\ \int \int e_{x_2} dt \end{bmatrix} \] (63)

and in Laplace domain

\[ u(s) = k_1 e + k_2 \left( \frac{x_{2r} - x_2}{s} \right) + k_3 \left( \frac{x_{2r} - x_2}{s} \right) + k_4 \left( \frac{x_{2r} - x_2}{s^2} \right) + k_5 \left( \frac{x_{2r} - x_2}{s^3} \right) + k_6 \left( \frac{x_{2r} - x_2}{s^4} \right) \] (64)

### 6.6.1 Parallel controller

\[ u = \left( k_1 + \frac{k_3}{s} + \frac{k_5}{s^2} \right) (y_r - y) + \left( k_2 + \frac{k_4}{s} + \frac{k_6}{s^2} \right) (x_{2r} - x_2) \] (65)

### 6.6.2 Cascade controller – the PI²V configuration

\[ u = \frac{k_2 s^2 + k_4 s + k_6}{s^2} \left( x_{2r} - x_2 \right) + \frac{k_1 s^2 + k_3 s + k_5}{k_2 s^2 + k_4 s + k_6} (y_r - y) \] (66)

This is called the PI²V configuration (Proportional feedback in position loop and proportional + double integral feedback in the velocity loop) as there is almost proportional feedback (Lead-Lag) in the position loop, on \( y \), and then in the velocity loop, on \( x_2 \), there is proportional and two integrals feedback.

### 6.6.3 One block output controller

\[ u(s) = \left( k_1 + \frac{k_3}{s} + \frac{k_5}{s^2} \right) (y_r - y) + \left( k_2 + \frac{k_4}{s} + \frac{k_6}{s^2} \right) (x_{2r} - x_2) \]

\[ C(s) = \frac{u(s)}{e(s)} = \left( k_1 + \frac{k_3}{s} + \frac{k_5}{s^2} \right) + \left( k_2 + \frac{k_4}{s} + \frac{k_6}{s^2} \right) \frac{b_2 s - a_2 b}{b_1 s + (b_2 + a_1 b_1)} \] (67)

This is the PFD controller.

### 6.6.4 Discussion

1. Remarks in section 6.1.4 apply here mutatis mutandis.

### 6.7 Summary

This section presented the family of the generalized PID controllers for 2\(^{nd}\) order systems. The following tables summarize the structure of the controllers in the different architectures.
Table 2 presents the family of generalized PID controllers for 2nd order systems in the parallel architecture that are able to drive the tracking error to zero for up to constant acceleration input and disturbance. Formally, if all possible parallel configurations are enumerated then there are three more parallel structures as detailed in Table 3. However these additional structures are equivalent to the respective structures in Table 2 as detailed in the rightmost column. Therefore these configurations are not considered in the following.

<table>
<thead>
<tr>
<th>Generalized PID controller - Parallel architecture (Figure 1)</th>
<th>$C_{x1}$</th>
<th>$C_{x2}$</th>
<th>§</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>$k_1$</td>
<td>$k_2$</td>
<td>6.1</td>
</tr>
<tr>
<td>PID</td>
<td>$k_1+k_3/s$</td>
<td>$k_2$</td>
<td>6.2</td>
</tr>
<tr>
<td>PID-PIV</td>
<td>$k_1+k_3/s$</td>
<td>$k_2+k_4/s$</td>
<td>6.3</td>
</tr>
<tr>
<td>PI$^2$D</td>
<td>$k_1+k_3/s+k_4/s^2$</td>
<td>$k_2$</td>
<td>6.4</td>
</tr>
<tr>
<td>PI$^2$D-PIIV</td>
<td>$k_1+k_3/s+k_5/s^2$</td>
<td>$k_2+k_4/s$</td>
<td>6.5</td>
</tr>
<tr>
<td>PI$^2$D-PI$^2$V</td>
<td>$k_1+k_3/s+k_5/s^2$</td>
<td>$k_2+k_4/s+k_6/s^2$</td>
<td>6.6</td>
</tr>
</tbody>
</table>

Table 2. The structure of the parallel architecture controllers for 2nd order plant.

<table>
<thead>
<tr>
<th>Generalized PID controller - Parallel architecture (Figure 1)</th>
<th>$C_{x1}$</th>
<th>$C_{x2}$</th>
<th>§</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>$k_1$</td>
<td>$k_2+k_4/s$</td>
<td>6.2</td>
</tr>
<tr>
<td>PI$^2$D</td>
<td>$k_1$</td>
<td>$k_2+k_4/s+k_6/s^2$</td>
<td>6.4</td>
</tr>
<tr>
<td>PI$^2$D-PIIV</td>
<td>$k_1+k_3/s$</td>
<td>$k_2+k_4/s+k_6/s^2$</td>
<td>6.5</td>
</tr>
</tbody>
</table>

Table 3. The structure of the parallel architecture controllers for 2nd order plant.

Tables 4 and 5 present the family of generalized PID controllers for 2nd order systems in the cascade architecture and in the one block controller architecture, respectively, that are able to drive the tracking error to zero for up to constant acceleration input and disturbance.

<table>
<thead>
<tr>
<th>Generalized PID controller - Cascade architecture (Figure 2)</th>
<th>$C_p$ (position-outer loop)</th>
<th>$C_v$ (velocity-inner loop)</th>
<th>§</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>$k_1$</td>
<td>$k_1/k_2$</td>
<td>6.1</td>
</tr>
<tr>
<td>PID</td>
<td>$(k_1s+k_3)/k_2/s$</td>
<td>$k_2$</td>
<td>6.2</td>
</tr>
<tr>
<td>PIV</td>
<td>$(k_1s+k_3)/(k_3s+k_4)$</td>
<td>$(k_3s+k_4)/k_2/s$</td>
<td>6.3</td>
</tr>
<tr>
<td>PI$^2$D</td>
<td>$(k_1s^2+k_3s+k_4)/k_2/s^2$</td>
<td>$k_2$</td>
<td>6.4</td>
</tr>
<tr>
<td>PI$^2$D-PIV</td>
<td>$(k_1s^2+k_3s+k_5)/(k_3s+k_4)$</td>
<td>$(k_3s+k_4)/s$</td>
<td>6.5</td>
</tr>
<tr>
<td>PI$^2$D-PI$^2$V</td>
<td>$(k_1s^2+k_3s+k_5)/(k_3s^2+k_4s+k_6)$</td>
<td>$(k_3s^2+k_4s+k_6)/s^2$</td>
<td>6.6</td>
</tr>
</tbody>
</table>

Table 4. The structure of the cascade architecture controllers for 2nd order plant.
7. Reference trajectory generator

The reference trajectory generator encapsulates the required closed loop behavior as stated by the system specification-requirements. There can be two cases: the trajectory is either unknown or known in advance. The former case gives the well known pre-filter that creates the feed-forward as well. In the second case, for example, minimum time trajectories for limited acceleration or jerk, minimum acceleration or jerk energy trajectories, or any other profile can be required. Both cases are presented in (Leonhard, 1996, pp. 80, 347, 363-364, 367) and in many other publications.

8. Discussion

In this chapter the generalized PID controllers for 1st and 2nd order system that are able to drive the tracking error to zero for up to second order polynomials inputs and disturbances have been derived. This presented in detail a methodology to derive additional members of the family of generalized PID controllers for high order system (Rusnak, 1999) and high order polynomial inputs and disturbances. These are the $\text{PI}^m\text{D}^{m-1}$ controllers.

Following the theory and the author's experience the full state feedback, especially the cascade architecture, Figure 2, is preferable over the one block controller, Figure 3. This may come at the expense of higher cost. However in modern digital control loop that are using absolute or incremental encoders the position and velocity information is derived at the same cost.

The motion control engineers prefer the cascade controller because of implementation and tuning easiness. The most important feature is that in the cascade architecture the feedback loop can be tuned sequentially. That is, start with the velocity-inner loop, that is usually high bandwidth, and then to proceed to the position-outer loop. The same apply to higher order generalized PID controllers.

9. Conclusions

By the use of LQR theory we formulated a control-tracking problem and showed those cases when their solution gives members of the $\text{PI}^m\text{D}^{m-1}$ family of controllers. This way heuristics are avoided and a systematic approach to explanation for the excellent performance of the PID controllers is given. The well known one block PID controller architecture is optimal for Linear Quadratic Tracking problem of 2nd order systems with no zero or stable zero.
10. References


This book discusses the theory, application, and practice of PID control technology. It is designed for engineers, researchers, students of process control, and industry professionals. It will also be of interest for those seeking an overview of the subject of green automation who need to procure single loop and multi-loop PID controllers and who aim for an exceptional, stable, and robust closed-loop performance through process automation. Process modeling, controller design, and analyses using conventional and heuristic schemes are explained through different applications here. The readers should have primary knowledge of transfer functions, poles, zeros, regulation concepts, and background. The following sections are covered: The Theory of PID Controllers and their Design Methods, Tuning Criteria, Multivariable Systems: Automatic Tuning and Adaptation, Intelligent PID Control, Discrete, Intelligent PID Controller, Fractional Order PID Controllers, Extended Applications of PID, and Practical Applications. A wide variety of researchers and engineers seeking methods of designing and analyzing controllers will create a heavy demand for this book: interdisciplinary researchers, real time process developers, control engineers, instrument technicians, and many more entities that are recognizing the value of shifting to PID controller procurement.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:
