Fuzzy PID Controllers
for Industrial Applications

G. Ron Chen

Lecture for EE 6452
City University of Hong Kong

Summary

- Proportional-Integral-Derivative (PID) controllers are the most widely used controllers in industries today
- **Statistics:** > 90% controllers in industries are PID or PID-type controllers (the rest are programmable logical controllers (PLC))
- **Merits** of PID controllers: simple, cheap, reliable, and effective
- For lower-order linear time-invariant systems and processes, PID controllers have good set-point tracking performance with guaranteed stability
- Fuzzy logic provides a certain level of artificial intelligence to the conventional PID controllers
- **Fuzzy PID** controllers have self-tuning ability and on-line adaptation to nonlinear, time-varying, and uncertain systems
- Fuzzy PID controllers provide a promising option for industrial applications with many desirable features
Outline of the Presentation

- Overview of the Fuzzy Logic Technology
- Overview of Conventional PID Controllers
- Introduction to Fuzzy PID Controllers
- Some Successful Examples of Applications
- Concluding Remarks
Overview of the Fuzzy Logic Technology

Closed-Loop Set-Point Tracking System

Consider the typical set-point tracking system:

\[ r(t) - y(t) \rightarrow 0 \quad (t \rightarrow \infty) \]

**Objective:**

\[ e(t) := r(t) - y(t) \rightarrow 0 \quad (t \rightarrow \infty) \]

**Approach:** Design a fuzzy logic controller (FLC)

![Figure 1](image1.png) A typical closed-loop set-point tracking system

![Figure 2](image2.png) General structure of a fuzzy logic controller
(i) If $e > 0$ then

$$e = r - y > 0 \quad \text{or} \quad r > y$$

the output $y$ is at position $a$ or $d$

(ii) Furthermore, if $\dot{e} < 0$ then

$$\dot{e} = \dot{r} - \dot{y} = 0 - \dot{y} \quad \text{or} \quad \dot{y} > 0$$

(iii) Therefore, the output $y$ is at position $a$

Fuzzy Logic Rule Base:

$R^1$: IF $e > 0$ AND $\dot{e} < 0$ THEN $u(t+) = u(t)$

$R^2$: IF $e < 0$ AND $\dot{e} < 0$ THEN $u(t+) = -u(t)$

$R^3$: IF $e < 0$ AND $\dot{e} > 0$ THEN $u(t+) = u(t)$

$R^4$: IF $e > 0$ AND $\dot{e} > 0$ THEN $u(t+) = -u(t)$
Fuzzy Controller Design

A. Fuzzification

**Purpose:** Enable the input physical signal to use the **rule base**

**Approach:** Use **membership functions**

![Figure 4](image-url) Four membership functions for signals $e$ and $\dot{e}$

B. Programmable Rule Base

$R^1$: IF $e = PL$ AND $\dot{e} < 0$ THEN $u(t^+) = \mu_{PL}(e) \cdot u(t)$

$R^2$: IF $e = PS$ AND $\dot{e} < 0$ THEN $u(t^+) = (1 - \mu_{PS}(e)) \cdot u(t)$

$R^3$: IF $e = NL$ AND $\dot{e} < 0$ THEN $u(t^+) = -\mu_{NL}(e) \cdot u(t)$

$R^4$: IF $e = NS$ AND $\dot{e} < 0$ THEN $u(t^+) = -(1 - \mu_{NS}(e)) \cdot u(t)$

$R^5$: IF $e = NL$ AND $\dot{e} > 0$ THEN $u(t^+) = \mu_{NL}(e) \cdot u(t)$

$R^6$: IF $e = NS$ AND $\dot{e} > 0$ THEN $u(t^+) = (1 - \mu_{NS}(e)) \cdot u(t)$

$R^7$: IF $e = PL$ AND $\dot{e} > 0$ THEN $u(t^+) = -\mu_{PL}(e) \cdot u(t)$

$R^8$: IF $e = PS$ AND $\dot{e} > 0$ THEN $u(t+1) = -(1 - \mu_{PS}(e)) \cdot u(t)$
To implement the FLC on a digital computer:

\[ u(t) = u(kT) \quad \text{and} \quad u(t+) = u((k+1)T) \]

where \( T \) is the sampling time.

\( R^1: \) If \( e(kT) = \text{PL} \) and \( \dot{e}(kT) < 0 \)

\[ \text{THEN } u((k+1)T) = \mu_{\text{PL}}(e(kT)) \cdot u(kT) \]

\( R^2: \) If \( e(kT) = \text{PS} \) and \( \dot{e}(kT) < 0 \)

\[ \text{THEN } u((k+1)T) = (1 - \mu_{\text{PS}}(e(kT))) \cdot u(kT) \]

\( R^3: \) If \( e(kT) = \text{NL} \) and \( \dot{e}(kT) < 0 \)

\[ \text{THEN } u((k+1)T) = -\mu_{\text{NL}}(e(kT)) \cdot u(kT) \]

\( R^4: \) If \( e(kT) = \text{NS} \) and \( \dot{e}(kT) < 0 \)

\[ \text{THEN } u((k+1)T) = -(1 - \mu_{\text{NS}}(e(kT))) \cdot u(kT) \]

\( R^5: \) If \( e(kT) = \text{NL} \) and \( \dot{e}(kT) > 0 \)

\[ \text{THEN } u((k+1)T) = \mu_{\text{NL}}(e(kT)) \cdot u(kT) \]

\( R^6: \) If \( e(kT) = \text{NS} \) and \( \dot{e}(kT) > 0 \)

\[ \text{THEN } u((k+1)T) = (1 - \mu_{\text{NS}}(e(kT))) \cdot u(kT) \]

\( R^7: \) If \( e(kT) = \text{PL} \) and \( \dot{e}(kT) > 0 \)

\[ \text{THEN } u((k+1)T) = -\mu_{\text{PL}}(e(kT)) \cdot u(kT) \]

\( R^8: \) If \( e(kT) = \text{PS} \) and \( \dot{e}(kT) > 0 \)

\[ \text{THEN } u((k+1)T) = -(1 - \mu_{\text{PS}}(e(kT))) \cdot u(kT) \]

where \( \dot{e}(kT) \approx \frac{1}{T}[e(kT) - e((k-1)T)] \), with initial conditions

\[ y(0) = 0, \quad e(-T) = e(0) = r - y(0), \quad \dot{e}(0) = \frac{1}{T}[e(0) - e(-T)] = 0 \]
C. Defuzzification

Select membership functions for the different control outputs from the rule base

Then, the overall control signal, \( u \), is generated by a weighted average formula:

\[
   u((k+1)T) = \frac{\sum_{i=1}^{N} \mu_i u_i(kT)}{\sum_{i=1}^{N} \mu_i}, \quad (\mu_i \geq 0, \sum_{i=1}^{N} \mu_i > 0)
\]

where control outputs \( u_i(kT), i = 1, ..., N=8 \) are from the rule base.
Overview of Conventional PID Controllers

In the \textbf{time domain}:

(i) \textbf{P-controller} \quad u(t) = K_P e(t)

(ii) \textbf{I-controller} \quad u(t) = K_I \int_0^t e(\tau) \, d\tau

(iii) \textbf{D-controller} \quad u(t) = K_D \frac{d}{dt} e(t)

Control gains, $K_P$, $K_I$, and $K_D$, are constants to be determined in the design for set-point tracking and stability consideration.

\begin{figure}[h]
\centering
\begin{subfigure}{0.5\textwidth}
\centering
\begin{tikzpicture}
\node [input] (r) {r};
\node [node] (e) at (1,0) {e};
\node [block] (kp) at (2.5,0) {$K_P$};
\node [input] (r) at (4,0) {y};
\node [block] (system) at (5,0) {system};
\node [block] (u) at (4.5,0) {u};
\draw (r) -- (e);
\draw (e) -- (kp);
\draw (kp) -- (u);
\draw (u) -- (system);
\end{tikzpicture}
\caption{(a) Proportional controller}
\end{subfigure}
\begin{subfigure}{0.5\textwidth}
\centering
\begin{tikzpicture}
\node [input] (r) {r};
\node [node] (e) at (1,0) {e};
\node [block] (ki) at (2.5,0) {$K_I \int_0^t$};
\node [input] (r) at (4,0) {y};
\node [block] (system) at (5,0) {system};
\node [block] (u) at (4.5,0) {u};
\draw (r) -- (e);
\draw (e) -- (ki);
\draw (ki) -- (u);
\draw (u) -- (system);
\end{tikzpicture}
\caption{(b) Integral controller}
\end{subfigure}
\begin{subfigure}{0.5\textwidth}
\centering
\begin{tikzpicture}
\node [input] (r) {r};
\node [node] (e) at (1,0) {e};
\node [block] (kd) at (2.5,0) {$K_D \frac{d}{dt}$};
\node [input] (r) at (4,0) {y};
\node [block] (system) at (5,0) {system};
\node [block] (u) at (4.5,0) {u};
\draw (r) -- (e);
\draw (e) -- (kd);
\draw (kd) -- (u);
\draw (u) -- (system);
\end{tikzpicture}
\caption{(c) Derivative controller}
\end{subfigure}
\caption{Conventional PID controllers}
\end{figure}
In the **frequency domain**

(i) **P-controller** \( U(s) = K_P E(s) \)

(ii) **I-controller** \( U(s) = \frac{K_I}{s} E(s) \)

(iii) **D-controller** \( U(s) = K_D s E(s) \)

Use Laplace transform \( L\{\cdot\} \) for continuous-time signals:

\[
U(s) = L\{ u(t) \} \quad \text{and} \quad E(s) = L\{ e(t) \}
\]

*Use z-transform \( Z\{\cdot\} \) for discrete-time signals.*
Figure 7 Some typical combination of P, I, D controllers.
A. Discretization of Conventional PID Controllers

First, digitize the conventional analog PID controllers by

\[ S = \frac{2}{T} \frac{z - 1}{z + 1} \]

where \( T > 0 \) is the sampling time.

For the PI controller:

\[ u(nT) = u(nT-T) + T \Delta u(nT) \]

\[ \Delta u(nT) = \tilde{K}_p v(nT) + \frac{\tilde{K}_I}{T} e(nT) \]

**Figure 8.** The digital PI controller
Similarly, for the **PD controller**:

\[
\Delta u(nT) = \tilde{K}_p \, d(nT) + \tilde{K}_D \, v(nT)
\]

where

\[
d(nT) = \frac{e(nT) + e(nT - T)}{T} \\
v(nT) = \frac{e(nT) - e(nT - T)}{T}
\]

**Figure 9.** The digital PD controller
B. Example: Designing the Fuzzy PI Controller

![Diagram of Fuzzy PI Controller]

**Figure 10.** The Fuzzy PI control system

**Fuzzification** and **Defuzzification** fuzzy membership functions

![Output membership functions]

**Figure 11.** Output membership functions

![Input membership functions]

**Figure 12.** Input membership functions
\[ \Delta u_{PI}(nT) = \begin{cases} \frac{L[K_v d(nT) + K_p e_v(nT)]}{2(2L - K_v d(nT))}, & \text{in IC1, IC2, IC5, IC6} \\ \frac{L[K_v d(nT) + K_p e_v(nT)]}{2(2L - K_v e_v(nT))}, & \text{in IC3, IC4, IC7, IC8} \\ \frac{1}{2}[L + K_p e_v(nT)], & \text{in IC9, IC10} \\ \frac{1}{2}[L + K_v d(nT)], & \text{in IC11, IC12} \\ \frac{1}{2}[-L + K_v e_v(nT)], & \text{in IC13, IC14} \\ \frac{1}{2}[-L + K_v d(nT)], & \text{in IC15, IC16} \\ 0, & \text{in IC18, IC20} \\ -L, & \text{in IC17} \\ L, & \text{in IC19} \end{cases} \]

Note: Rigorous stability can be guaranteed
Some Successful Examples of Applications

There are many successful examples of fuzzy PID controllers.

1. Robotics

* Six-legged Insect Robot
(Dr. P. Soorakra, G. R. Chen, and students)
King Mongkut’s Institute of Technology, Bangkok, Thailand
*Multi-purpose Autonomous Robust Carrier for Hospitals (MARCH)*

(Dr. P. Sooraska, Prof. S. K. Tso, Dr. B. L. Luk, G. R. Chen, and students)

Centre for Intelligent Design, Automation and Manufacturing
City University of Hong Kong
Other Application Examples

A. **Process Control** (temperature, flow rate, injection)

B. **Vehicles Control** (parking, docking, backing truck-trailers)

C. **Electronic Appliances** (washer, dryer, camcorder-camera)

D. **Life-Support Systems** (space shuttle, submarine)

E. …… many more ……
(R4) IF \(d = d_p\) AND \(r = r_p\) THEN PD-output = \(o \cdot z\).

In these rules, \(e = sp - y\) is the error, where \(sp\) is the set-point, \(e = -\dot{y}\) is the rate of change of the error, “PD-output” is the fuzzy PD control output \(\Delta u_{PD}(n)\), “\(d \cdot p\)” means “error positive” and “\(o \cdot p\)” means “output positive,” etc. Finally, “AND” is the Zadeh’s logical “AND” defined by 
\[
\mu(A \land B) = \min\{\mu_A, \mu_B\}.
\]

Similarly, from the membership functions of the fuzzy I controller, the following control rules are used for the I component, where \(e(n - 1)\) is the delayed error signal.

(R5) IF \(e(n - 1) = e\cdot n\) AND \(r = r\cdot n\) THEN I-output = \(o \cdot z\).

(R6) IF \(e(n - 1) = e\cdot n\) AND \(r = r\cdot p\) THEN I-output = \(o \cdot z\).

(R7) IF \(e(n - 1) = e\cdot p\) AND \(r = r\cdot n\) THEN I-output = \(o \cdot z\).

(R8) IF \(e(n - 1) = e\cdot p\) AND \(r = r\cdot p\) THEN I-output = \(o \cdot p\).

In the above rules, “I-output” is the fuzzy I control output \(\Delta u_I(n)\), and the other terms are defined similarly to the PD component.

These eight rules altogether yield the control actions for the fuzzy PD-I control law.

3) Defuzzification: In the defuzzification step, for both fuzzy PD and I controllers, the commonly used “center of mass” formula is employed to defuzzify the incremental control of the fuzzy control law, (15) as shown in (17)

\[
\Delta u(n) = \frac{\sum\{\text{input membership value} \times \text{membership value of output}\}}{\sum \{\text{membership value of output}\}}.
\]

For the fuzzy PD controller, the value-ranges of the two inputs, the error and the rate of change of the error, are actually decomposed into ten adjacent input-combination (IC) regions, as shown in Fig. 7. The control rules for the fuzzy PD controller (R1)–(R4), with membership functions and IC regions together, are used to evaluate appropriate fuzzy control law’s for each region.

Now, by applying the values \(o \cdot p = L\), \(o \cdot n = -L\), \(o \cdot z = 0\), and the following straight line formulas obtained from the geometry of Fig. 7:

\[
d \cdot p = \frac{K_p d(n) + L}{2L},
\]
\[
d \cdot n = \frac{-K_p d(n) + L}{2L},
\]
\[
r \cdot p = \frac{K_d r(n) + L}{2L},
\]
\[
r \cdot n = \frac{-K_d r(n) + L}{2L},
\]

we obtain the following nine formulas for the ten IC regions:

\[
\Delta u_{PD}(n) = \frac{1}{2}[K_p d(n) - K_d r(n)] \quad \text{in IC A} \quad (18)
\]
\[
\Delta u_{PD}(n) = \frac{1}{2}[K_p d(n) - K_d r(n)] \quad \text{in IC B} \quad (19)
\]
\[
\Delta u_{PD}(n) = \frac{1}{2}[-K_d r(n) + L] \quad \text{in IC C} \quad (20)
\]
\[
\Delta u_{PD}(n) = \frac{1}{2}[K_p d(n) - L] \quad \text{in IC D} \quad (21)
\]
\[
\Delta u_{PD}(n) = \frac{1}{2}[-K_d r(n) - L] \quad \text{in IC E} \quad (22)
\]
\[
\Delta u_{PD}(n) = \frac{1}{2}[K_p d(n) + L] \quad \text{in IC F} \quad (23)
\]
\[
\Delta u_{PD}(n) = 0 \quad \text{in IC G, J} \quad (24)
\]
\[
\Delta u_{PD}(n) = -L \quad \text{in IC H} \quad (25)
\]
\[
\Delta u_{PD}(n) = L \quad \text{in IC I} \quad (26)
\]

Similarly, defuzzification of the fuzzy I controller follows the same procedure as described above for the PD component, except that the input signals in this case are different. The
the mass-center of the axis of rotation, and $g$ the gravity constant.

The actual data that we used in the experiment were:

$I_1 = 0.030$ (kgm$^2$), $I_2 = 0.004$ (kgm$^2$), $Mgl = 0.800$ (Nm), $K = 31.00$ (Nm/rad).

IV. EXPERIMENT AND RESULTS

In the experiment, the robot arm system was run using two control methods: using the conventional digital PID controller and the fuzzy PID controller, respectively, both based on the same configuration and under the same conditions. The conventional digital PID controller was unable to satisfactorily run the robot arm system (with a nonlinear load) no matter what combinations of control gains were used. Plots of the system output with the tuned PID gains that achieved the best results by trial-and-error are shown in Fig. 9, where and below $T = 0.001$ s. The fuzzy PID controller, in contrast, controlled the robot arm system with reasonably good results, obtained also by trial-and-error, as shown in Fig. 10. In this and the other fuzzily controlled simulation figures, $P = K_p$, $I = K_i$, $D = K_d$, and $K_{ud}$ and $K_{ub}$ are fuzzy gains defined in Section II-B (see Fig. 2). In Figs. 9 and 10, the robot arm was not connected to the spring, where the units used are voltage (1 V = 400 r/min) versus second.

Another comparison, at a different speed and with the spring added to the arm, is shown in Figs. 11 and 12, where the units used are voltage (1 V = 400 r/min) versus $t$s in Fig. 11 while versus second in Fig. 12. In these experiments, parameters for the fuzzy PID control gains were found such that there
Applications of Fuzzy PID Controllers on Process Systems

Conventional Control With 2000 Data Points

Fuzzy Control For 2000 Data Points
Concluding Remarks

• Fuzzy logic provides a certain level of artificial intelligence to the conventional PID controllers, leading to the effective fuzzy PID controllers

• Fuzzy PID controllers are easy to use (plug-and-play)

• Fuzzy PID controllers can be implemented by both
  o software (matured)
  o hardware (pre-matured)

• Fuzzy PID controllers have strong self-tuning ability and on-line adaptation to nonlinear, time-varying, and uncertain systems.

• Fuzzy PID controllers have a promising future for various industrial applications