Permissive or Blocking Pilot Protection Schemes? How to Have It Both Ways

Bogdan Kasztenny
Mangapathirao V. Mynam
Normann Fischer
Armando Guzmán
Schweitzer Engineering Laboratories, Inc.

Overview

- Pilot scheme fundamentals
- Pilot logic – built-in or freely programmed?
- Unified pilot logic
- Application considerations
- Crossover permissive-blocking scheme
**Pilot scheme fundamentals**

- Pilot scheme trips instantaneously and selectively by comparing fault direction at all line terminals
- Single-bit channel conveys fault direction
- Two basic scheme types are in use
  - **Permissive**
    Trip if the relay and all remote relays see the fault as forward
  - **Blocking**
    Trip if the relay sees the fault as forward and none of the remote relays see the fault as reverse

---

**Permissive pilot logic**

[Diagram showing PILOT logic with labels for PILOTF, PILOT, PILOTX, PILOTX, CRPU, CRDO, and POTT, with notes for each label:]

- PILOTF: user-selected forward-looking overreaching protection elements
- PILOT: reverse-looking protection elements matching PILOTF selection
- CRPU/CRDO: current-reversal blocking coordination timer
- PILOTX: pilot signal transmitted (permission)
- PILOTX: pilot signal received (permission)
- POTT: POTT scheme operation (trip)
**Blocking pilot logic**

PILOTF and PILOTR
forward- and reverse-looking protection elements

CRPU/CRDO
current-reversal blocking coordination timer

PILOTX and PILOTRX
pilot signal transmitted and received (blocking)

COORD
DCB coordination timer

DCB
DCB scheme operation (trip)

DCBSTOP
Stop signal for ON/OFF channels

---

**Dependability and security**

- Permissive and blocking schemes
  - Perform almost identically under normal conditions
  - Respond differently only under failure modes
- Failure modes
  - Failed equipment
    - (relay, channel, PT, test switch left open)
  - Unusual or stressed system condition
    - (weak infeed, open breaker, settings issue)
- Permissive schemes may miss faults
- Blocking schemes may overtrip
**Pilot logic – built-in or freely programmed?**

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Built-In</th>
<th>Freely Programmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multivendor consistency</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Longevity throughout installation lifecycle</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Familiarity and ease of testing and troubleshooting</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Initial engineering and testing</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Maximizing product capability</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Single-pole tripping</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

**Can we unify permissive and blocking logic?**
Unified pilot logic

EPILOT
enable setting (POTT or DCB)

PILOT
user-selectable list of forward-looking protection elements (level/zone 2)

PILOT
relay-matched list of reverse-looking protection elements (level/zone 3, 4, or 5)

PIOTX and DCBSTOP
output bits freely configurable to drive any relay outputs

PIOTRX
programmable equation to receive pilot signal from any relay input

POTT / DCB
POTT or DCB operation signal

Three-terminal POTT with test bit

EPILOT = POTT

From Relay 2
From Relay 3
TEST

PIOTR
POTX
POTT / DCB
DCBSTOP
PIOTRX
To Relay 2
To Relay 3
Two-terminal DCB with redundant channels

PUTT application
Separating permissive bits

POTT channel failure considerations

Channel Failure Key Logic ("DCUB")

Channel Failure Echo Logic
POTT schemes just got much faster!

- 09/12/2019 fault on a 345 kV, 109 mi line
- TD32 asserted in 1 ms (PILOTEX sent)
- PILOTRX received in 2.6 ms
- POTT tripped in 2.6 ms
- TW87 tripped in 1 ms
- Fault cleared in 25 ms

Crossover permissive-blocking scheme
Crossover scheme benefits

- Simple engineering and settings
  - No need for open-breaker echo logic (DCB trips)
  - No need for weak-infeed logic (DCB trips)
  - No need for channel failure or echo logic (DCB trips)
  - Generous DCB coordination time setting (DCB security)
- Fast operation (POTT trips)
- Dependable operation (DCB trips)
- Requires one more input, output, and channel bit
- Caution: reverse elements must be set carefully

Key takeaway

- If you are a permissive scheme user, consider the crossover scheme to simplify your application
- If you are a blocking scheme user, consider the crossover scheme for speed of operation

<table>
<thead>
<tr>
<th>Blocking Bit</th>
<th>Permissive Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>DCB trip</td>
</tr>
<tr>
<td>01</td>
<td>POTT trip (faster)</td>
</tr>
<tr>
<td>10</td>
<td>No trip (external fault)</td>
</tr>
</tbody>
</table>
Summary

- Programmable logic equations allow good compromise between using built-in logic and the need to customize
- Test bits, redundant channel logic, multiterminal line logic, and channel failure logic are best left to programmable logic equations
- Crossover permissive-blocking scheme offers good compromise between speed, security, and dependability

Backup slides for discussion
POTT open-breaker echo logic

ECHORX: freely programmable request to echo
ECBECCHO: freely programmable CB open, disconnect switch open, or stub-bus condition bit

Open-breaker echo applications

Three-Terminal Line Application

Channel Failure Echo Logic
POTT weak-infeed echo logic

- **ECHORX**: freely programmable request to echo
- **UV**: user-selectable abnormal voltage condition (27P, 59Q, 59G)

Diagram shows the logic flow with inputs and outputs labeled accordingly.