Pertemuan ke-14

Pengontrolan variabel analog menggunakan PLC: Algoritma PID

Garis Besar & Tujuan Sesi

- Memahami apa itu kontrol PID
- Mengetahui fungsi dari setiap istilah kontrol PID
- Bisa memilih kombinasi yang benar dari elemen kontrol PID untuk berbagai tujuan aplikasi kontrol proses
- Mengetahui fitur-fitur tambahan yang terinstall pada pengontrol untuk diimplementasikan dalam prakteknya

Apakah Kontrol PID itu?

- PID singkatan dari **P**roportional - **I**ntegrator - **D**erivative
- Juga dikenal sebagai kontrol **tiga-istilah**
- Diimplementasikan seperti program komputer sekarang
  - Pengontrol muncul dengan berbagai bentuk yang **berbeda**
  - Kontrol PID sering dikombinasikan dengan blok fungsi logik, sekuensial dan sederhana untuk membangun sistem otomasi yang kompleks

Mengapa harus Kontrol PID?

- Algoritma PID sederhana, mudah dimengerti dan relatif lebih mudah untuk di-tuning dibanding pengontrol yang lain
  - Telah menjadi peralatan standar ketika kontrol proses muncul pada tahun 1940
  - Dalam kontrol proses sekarang, lebih dari 90% kontrol loop adalah dari jenis PID

Konsep Utama (1)

- Aksi Proses
  - Menetapkan hubungan antara perubahan dalam **katup** (valve) dan perubahan dalam **pengukuran**

DIRECT	**Penambahan** posisi katup menyebabkan **penambahan** pengukuran
REVERSE	**Penambahan** posisi katup menyebabkan **penurunan** pengukuran

- Aksi pengontrol
  - Menetapkan hubungan antara perubahan dalam **variabel terukur** dan perubahan dalam **keluaran pengontrol**

DIRECT	**Penambahan** variabel terukur menyebabkan **penambahan** keluaran
REVERSE	**Penambahan** variabel terukur menyebabkan **penurunan** keluaran

Aksi pengontrol haruslah **berlawanan** dengan aksi proses

Konsep Utama (2)

- Algoritma PID tidak mengetahui keluaran yang benar untuk membawa variabel proses pada setpoint
  - Algoritma harus mempunyai pengukuran proses untuk berjalan
  - Ia hanya melanjutkan untuk menggerakkan keluaran pada arahnya yang harusnya menggerakkan proses menuju setpoint

- Algoritma PID harus ‘di-tuning’ untuk loop proes tertentu
  - Setiap istilah dari persamaan PID harus **dipahami**
  - Tuning **berdasarkan** pada dinamik dari respon proses

Konsep Utama (3)

Mode Manual & Auto

- Mode Manual
  - Operator manusia mengatur keluaran untuk mengoperasikan proses
  - Mode Manual sangat berguna jika terjadi kondisi yang tidak biasanya:
    - Start-up Plant
    - Shut-down Plant
    - Darurat
- Mode Auto
  - Algoritma kontrol memanipulasi keluaran untuk menahan pengukuran proses pada setpoint-nya
  - Seharusnya merupakan mode paling umum untuk operasi normal

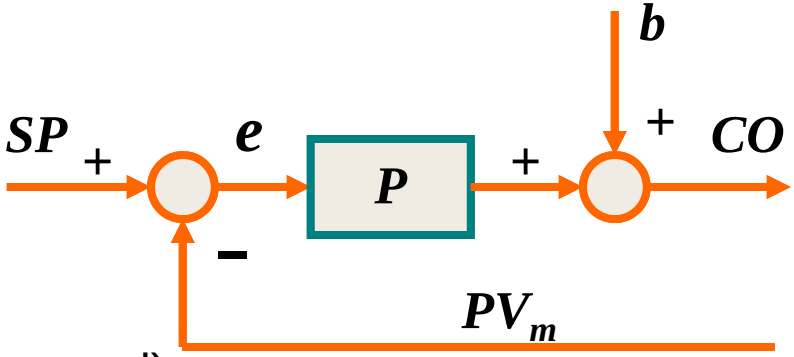
Algoritma Kontrol PID

- Terdiri dari tiga elemen:
  - **P**roportional – juga dikenal sebagai Gain Proporsional atau cukup Gain
  - **I**ntegral – juga dikenal sebagai Reset Otomatis atau cukup Reset
  - **D**erivative – juga dikenal sebagai Rate atau Pre-Act

- Tersedia dalam beberapa kombinasi dari elemen-elemen berikut:
  - Hanya Proporsional (P)
  - Proportional dan Integral (PI) (paling umum)
  - Proportional, Integral, dan Derivative (PID)
  - Proportional dan Derivative (PD)

Mode Proporsional (1)

$$CO = P \cdot e + b$$



P = gain proporsional (tidak berdimensi)

CO = pengontrol output (%)

b = bias (%) (dikenal juga sebagai reset manual)

e = ($SP - PV_m$) (%) → "aksi kebalikan", atau

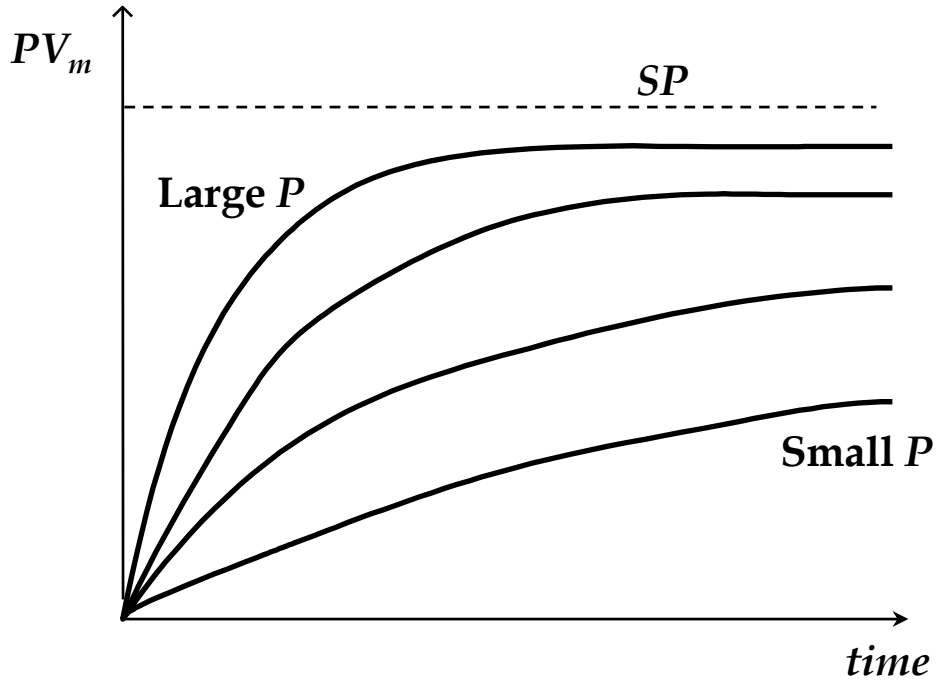
e = – ($SP - PV_m$) (%) → "aksi langsung"

- Beberapa pabrik menggunakan **Proportional Band** (***PB***) daripada gain proporsional
  - *PB* adalah % perubahan dalam input yang mana 100% perubahan terjadi dalam keluaran

$$PB = \frac{100}{P} \Rightarrow CO = P \cdot e + b = \frac{100}{PB} e + b$$

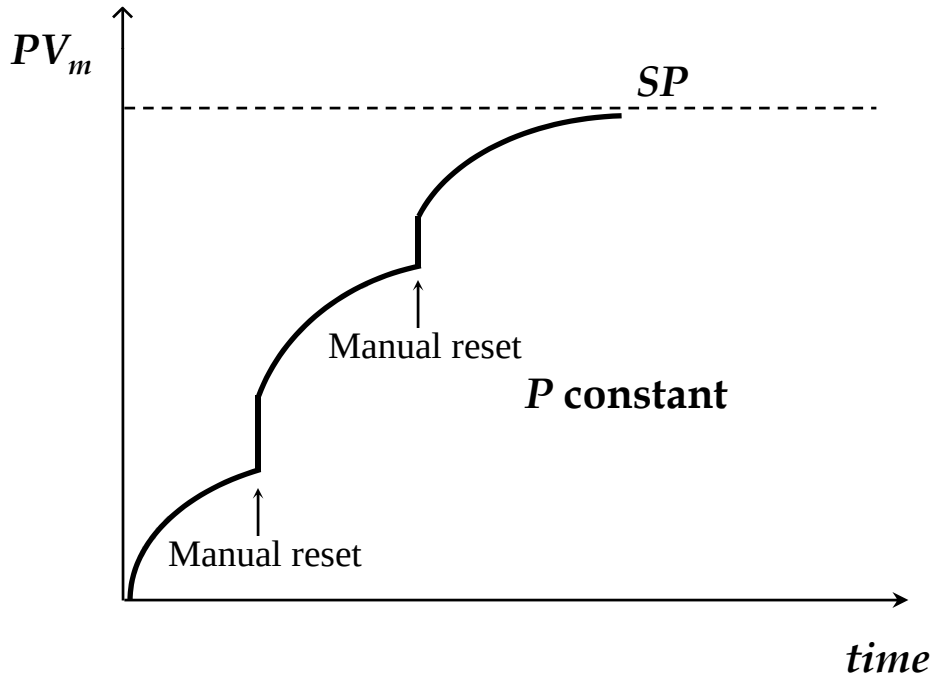
Mode Proporsional (2)

- Kontrol hanya-proporsional bisa menghasilkan **offset**
  - Offset bisa **dikurangi** dengan **meningkatkan gain** pengontrol (atau **menurunkan band proporsional**). Tapi satu tidak bisa membuat gain pengontrol besar sewenang-wenang karena **terlalu tinggi sebuah gain menyebabkan osilasi dan/atau ketidakstabilan**



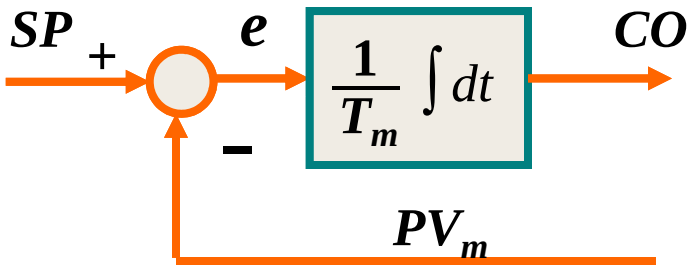
Mode Proporsional (2)

- Untuk **menghilangkan** offset, operator manusia harus me-“**reset**” pengontrol secara manual dengan mengatur harga reset (istilah “b”)



Mode Integral (Reset Otomatis) (1)

$$CO = \frac{1}{T_m}\int e\,dt$$



T_m = waktu integral (menit per repeat atau detik per repeat)

CO = pengontrol output (%)

e = ($SP – PV_m$) (%) → “aksi kelbalikan” atau

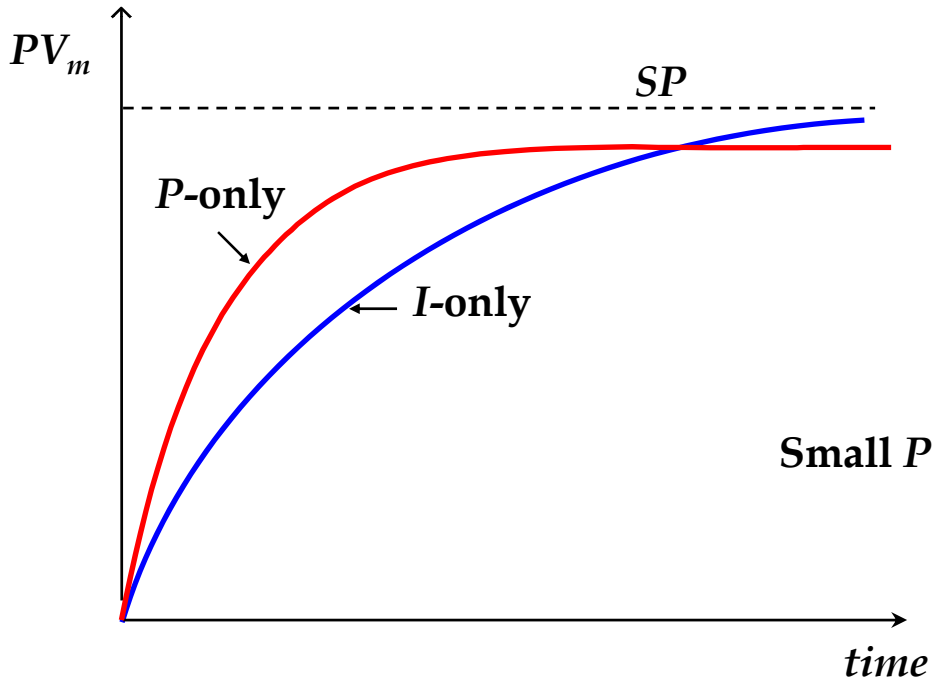
e = – (SP – PV_m) (%) → “aksi langsung”

- Beberapa pabrik menggunakan **repeat per menit** (atau **repeat per detik**) daripada menit per repeat (atau detik per repeat)
  - Repeat per menit (atau detik) adalah waktu yang diperlukan melakukan reset (atau integral) element untuk mengulangi (reset) aksi dari elemen proporsional

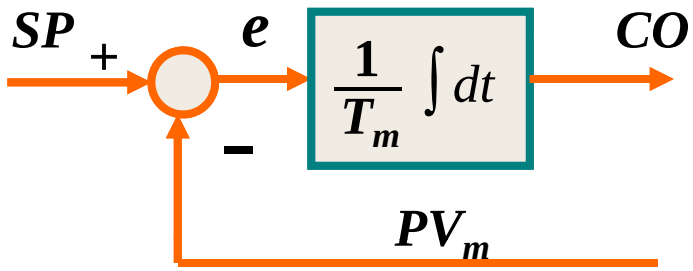
$$\text{repeat per menit } (T_r) = \frac{1}{\text{menit per repeat } (T_m)} \Rightarrow CO = T_r \int e\,dt$$

Mode Integral (Reset Otomatis) (2)

- Selama terdapat error, pengontrol akan merubah keluarannya; karenanya ia bisa mengendalikan error menjadi nol
- Kecepatan respon berkurang (dibanding mode P-only)



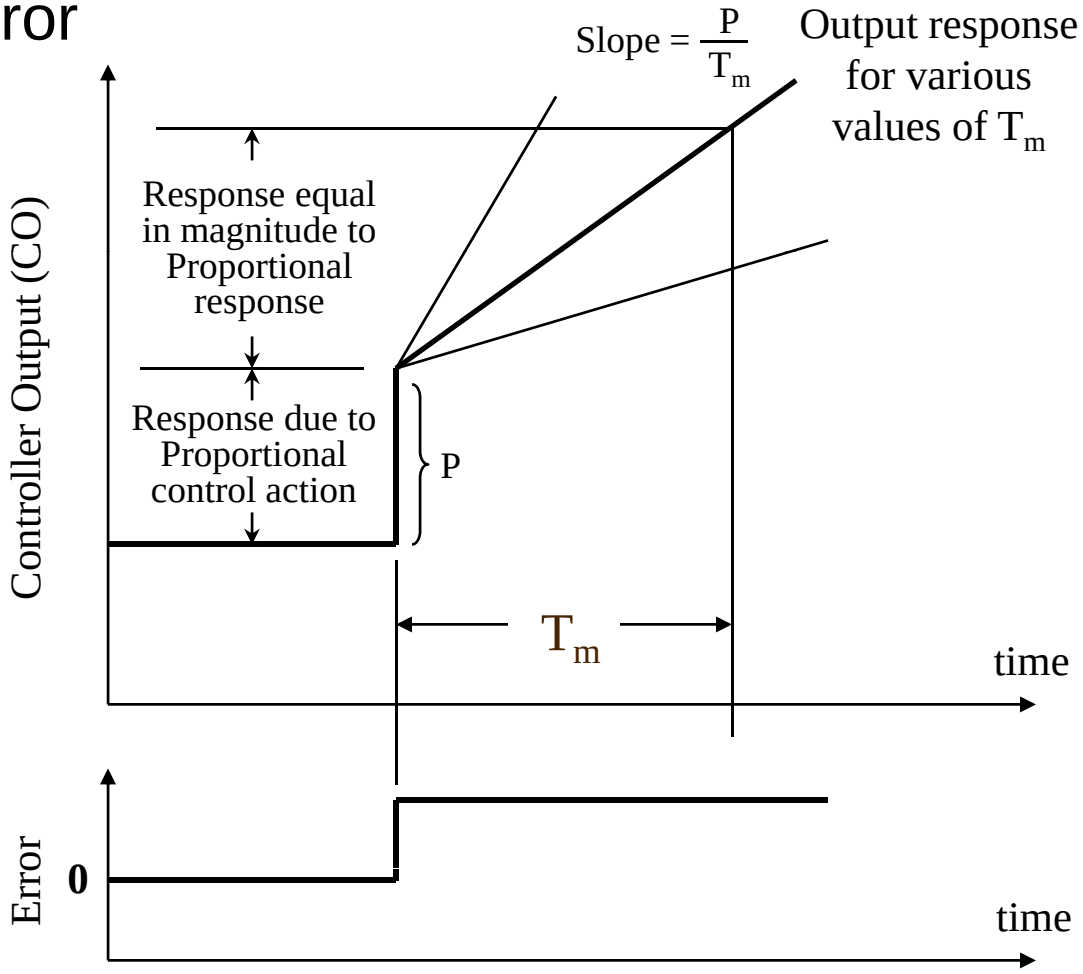
Mode Proporsional-Integral (1)



$$CO = P\left(e + \frac{1}{T_m}\int e\,dt\right)$$

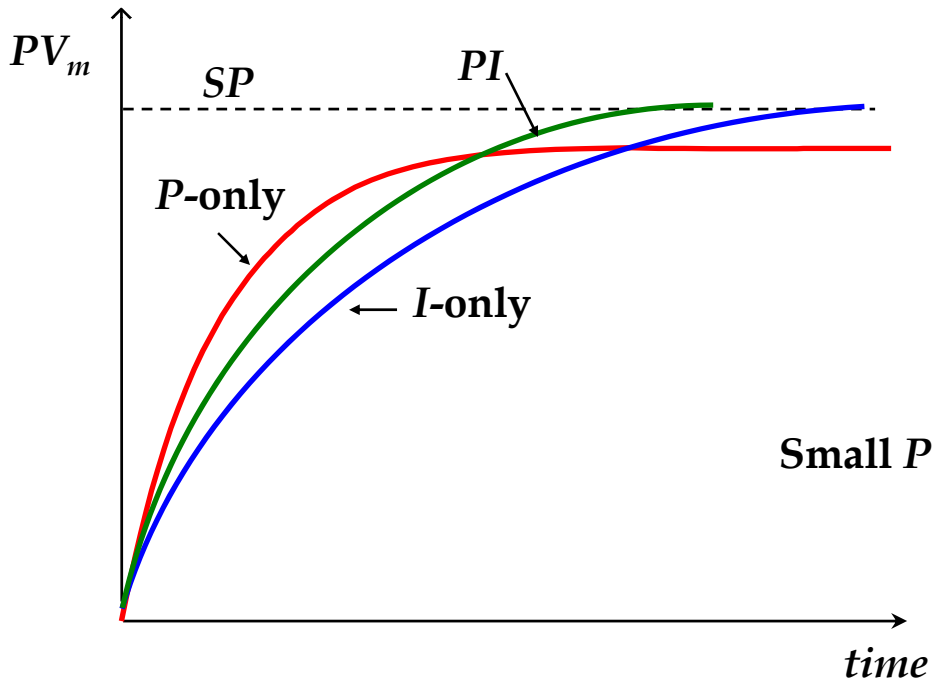
Mode Proporsional-Integral (2)

- Respon dari pengontrol PI terhadap perubahan langkah dalam error



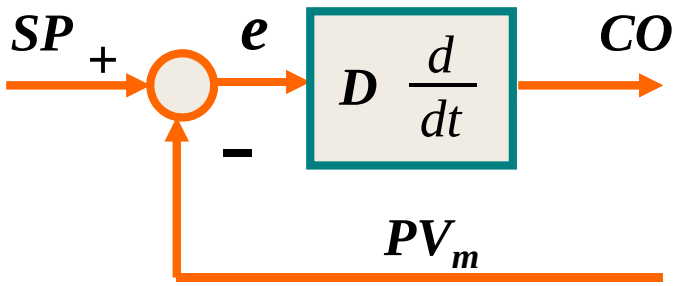
Mode Proporsional-Integral (3)

- Mengkombinasikan fitur terbaik dari mode proporsional dan integral
  - Offset proporsional **dihilangkan** dengan **sedikit kehilangan** kecepatan respon



Mode Derivatif (1)

$$CO = D\,\frac{de}{dt}$$



D = waktu derivatif (menit atau detik)

CO = pengontrol output (%)

e = ($SP - PV_m$) (%) → “reverse action” atau

e = – ($SP - PV_m$) (%) → “direct action”

- Kecepatan respons meningkat (dibandingkan mode P-only)
- Hipersensitif terhadap noise dan disturbances berfrekuensi-tinggi lainnya

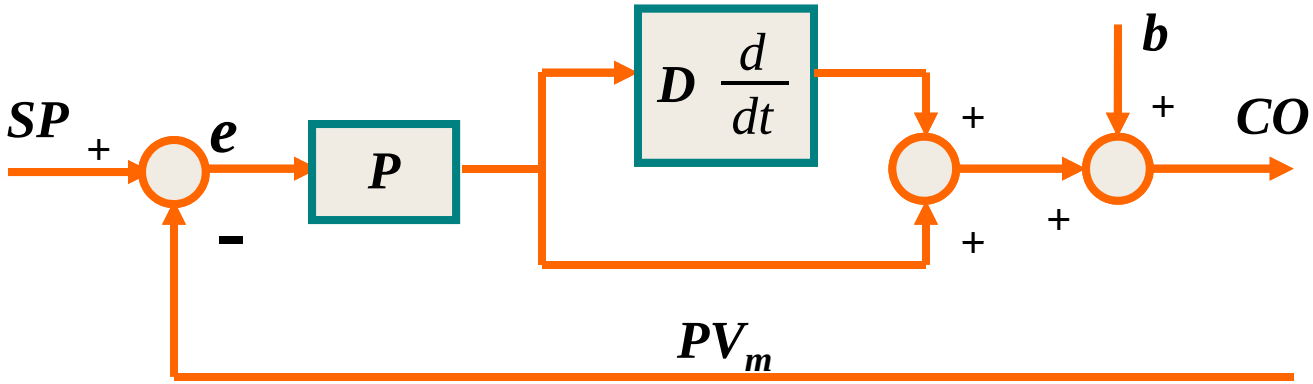
Mode Derivatif (2)

- Sinyal error steady-state, bagaimanapun, tidak dikenali oleh pengontrol ***D***, karena tanpa memperhatikan berapa besarnya error, **tingkat perubahannya adalah nol**.

  Karena itu, pengontrol hanya-derivatif tidak digunakan dalam praktek

- Biasanya mereka ditemukan dalam kombinasi dengan eleman kontrol lainnya, kebanyakan dalam kombinasi dengan kontrol proporsional

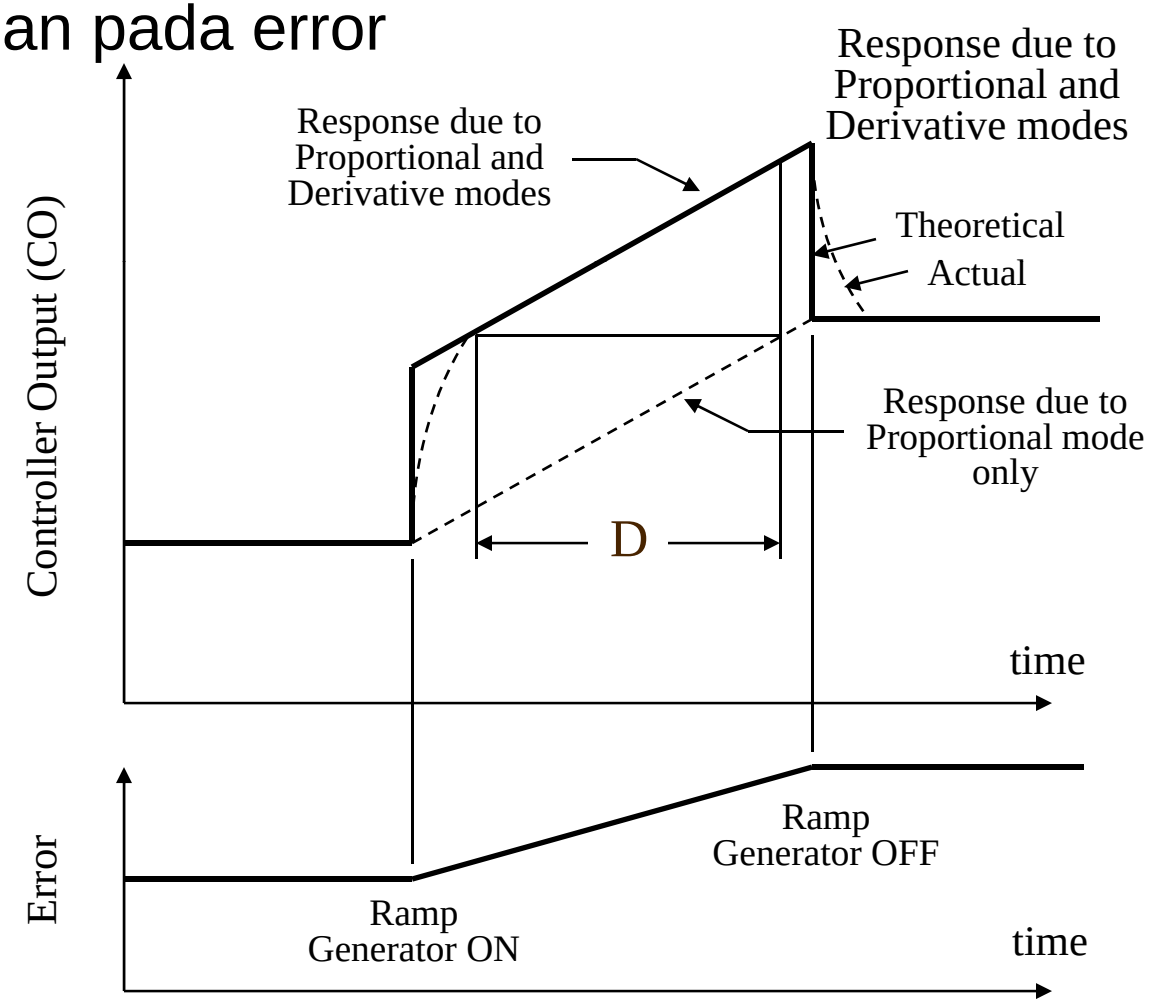
Mode Proporsional-Derivatif (1)



$$CO = P\left(e + D\frac{de}{dt}\right) + b$$

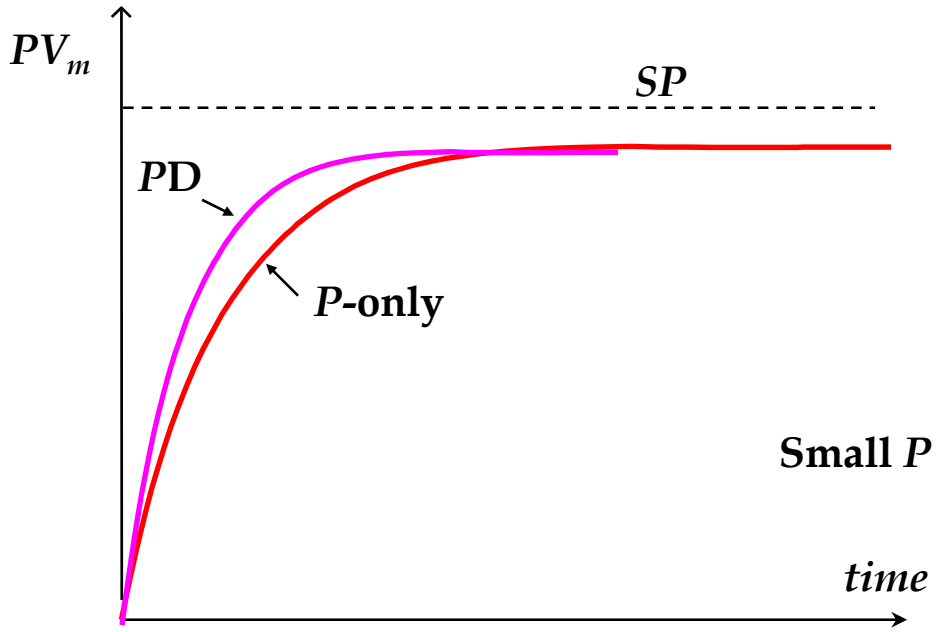
Mode Proporsional-Derivatif (2)

- Respons dari pengontrol PD untuk melandaikan perubahan pada error



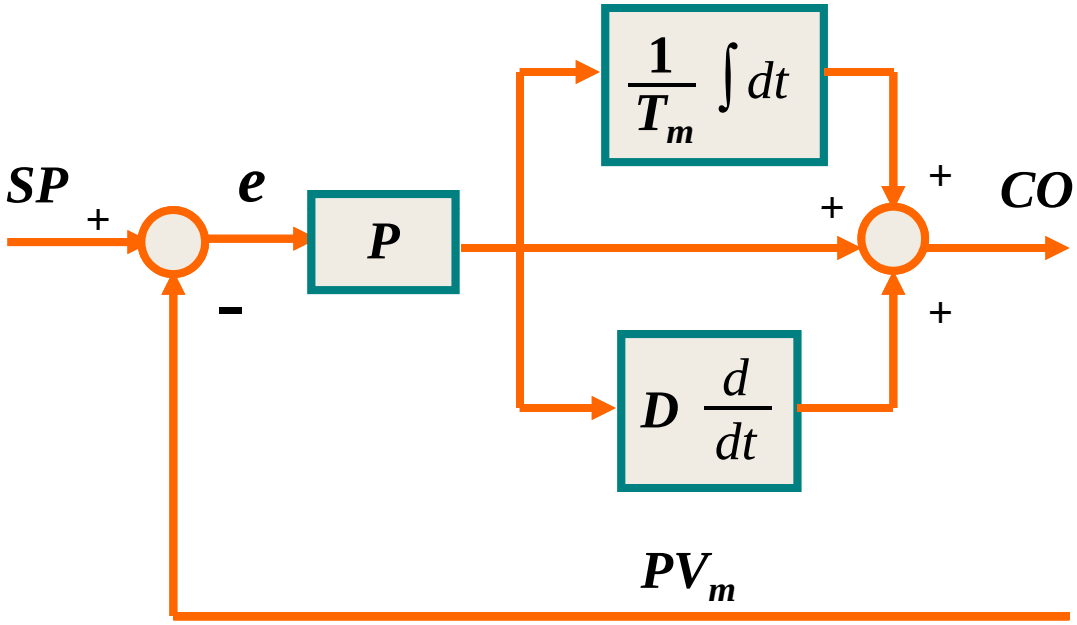
Mode Proporsional-Derivatif (3)

- Kontrol PD bisa menghasilkan sebuah **offset**
- Untuk menghindari offset proporsional, bias "***b***" seharusnya diatur ketika PV_m ada pada setpoint
- Biasanya ditemukan pada slow-response process kontrol, misalnya kontrol temperatur, pH, komposisi



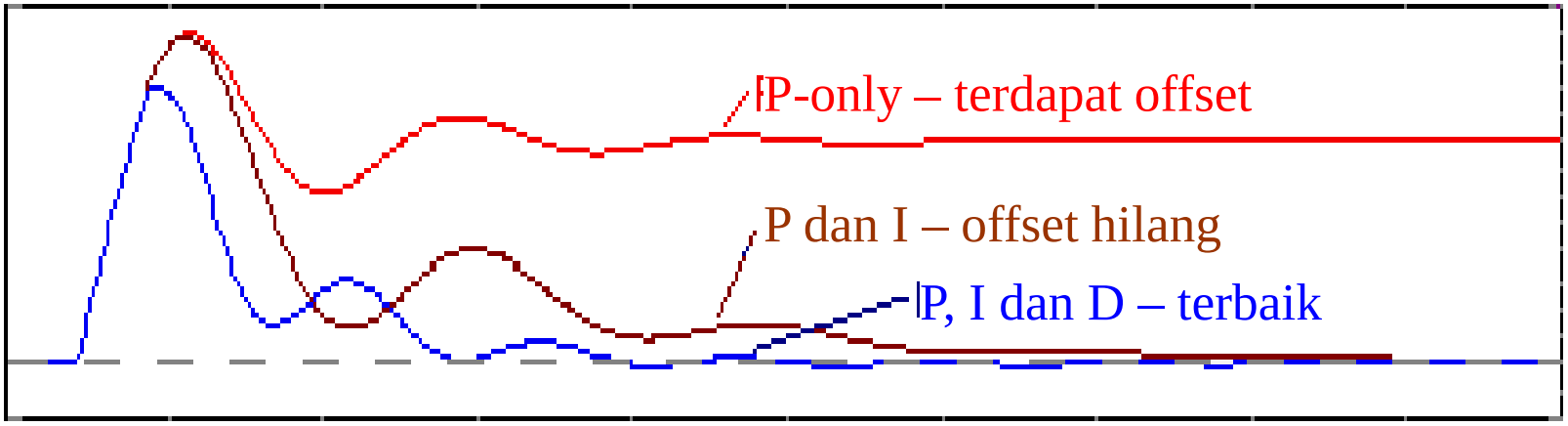
Mode Proporsional-Integral-Derivatif (1)

- Mengkombinasikan fitur terbaik dari istilah P, I, dan D



$$CO = P\left(e + \frac{1}{T_m}\int e\,dt + D\frac{de}{dt}\right)$$

Mode Proporsional-Integral-Derivatif (2)



Konsep Tambahan PID

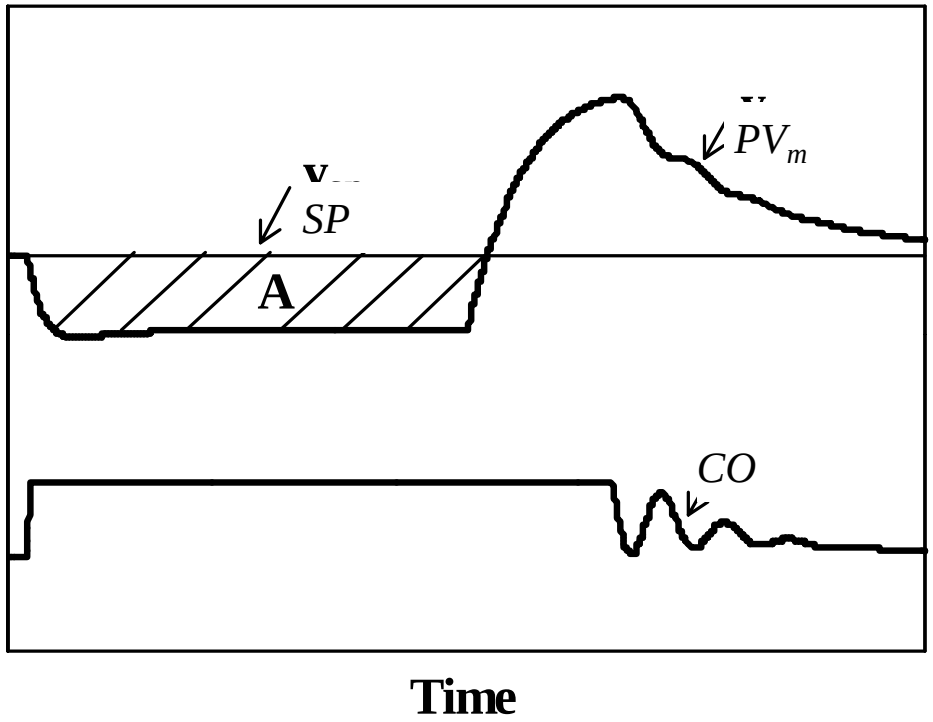
Algoritma PID Interaktif vs. Non-interaktif

- Mengacu pada interaksi antara reset dan istilah derivatif
- Juga dikenal sebagai ‘seri’ atau ‘parallel’
- Hampir semua pengontrol analog adalah interaktif
- Banyak pengontrol digital adalah non-interaktif, beberapa adalah interaktif
- Satu-satunya perbedaan adalah pada saat tuning pengontrol dengan derivatif

Reset Windup (1)

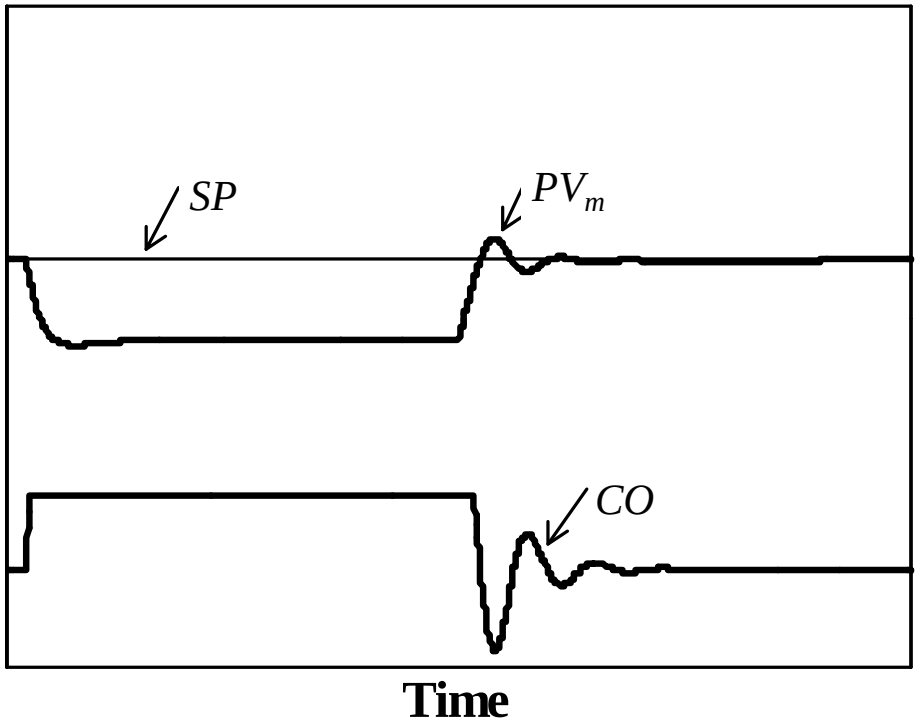
- Semua aktuator mempunyai batasan:
  - Contoh: sebuah motor mempunyai kecepatan terbatas, sebuah katup tidak bisa lebih dari terbuka penuh atau tertutup penuh
- Fenomena windup disebabkan oleh interaksi dari aksi integral dan saturasi
- Ketika ini terjadi loop umpan balik telah rusak dan sistem berjalan sebagai sebuah loop terbuka karena aktuator akan tetap pada batasnya secara bebas dari keluaran proses
  - Jika sebuah pengontrol dengan aksi integrasi digunakan, error akan berlanjut untuk diintegrasikan. Ini berarti integral istilah bisa menjadi sangat besar atau, biasa disebut, ia “winds up”
  - Adalah diperlukan bahwa error mempunyai tanda yang berlawanan untuk jangka waktu yang lama sebelum semuanya kembali normal
  - Konsekuensinya adalah sembarang pengontrol dengan aksi integral bisa memberikan transient yang besar ketika aktuator bersaturasi

Reset Windup (2)



- Ingat bahwa saturasi keluaran pengontrol menyebabkan area “A” diakumulasi oleh aksi integral
- Setelah disturbance kembali ke tingkat normalnya, keluaran pengontrol tetap disaturasi untuk beberapa periode waktu yang menyebabkan sebuah gangguan dalam PV_m

Anti-Reset Windup



- Ketika variabel termanipulasi mengalami saturasi, integral tidak diperbolehkan untuk diakumulasi
- Ketika kontrol telah kembali, pengontrol mengambil tindakan segera dan proses kembali dengan halus ke setpoint

Metode untuk Anti-Reset Windup

- Matikan integral ketika sebuah katup telah penuh atau sebuah kontrol loop tidak sedang digunakan.
- Awasi keluaran pengontrol agar lebih besar dari 0% dan kurang dari 100%.
- Berlakukan umpan balik reset internal
- Berlakukan umpan reset balik eksternal

Bumpless Transfer (1)

- Praktisnya semua pengontrol bisa berjalan dalam dua mode: **manual** atau **otomatis**

- Ketika sistem dalam mode manual, algoritma kontrol menghasilkan sinyal kontrol yang bisa berbeda dari sinyal kontrol yang dihasilkan secara manual, atau sebaliknya.

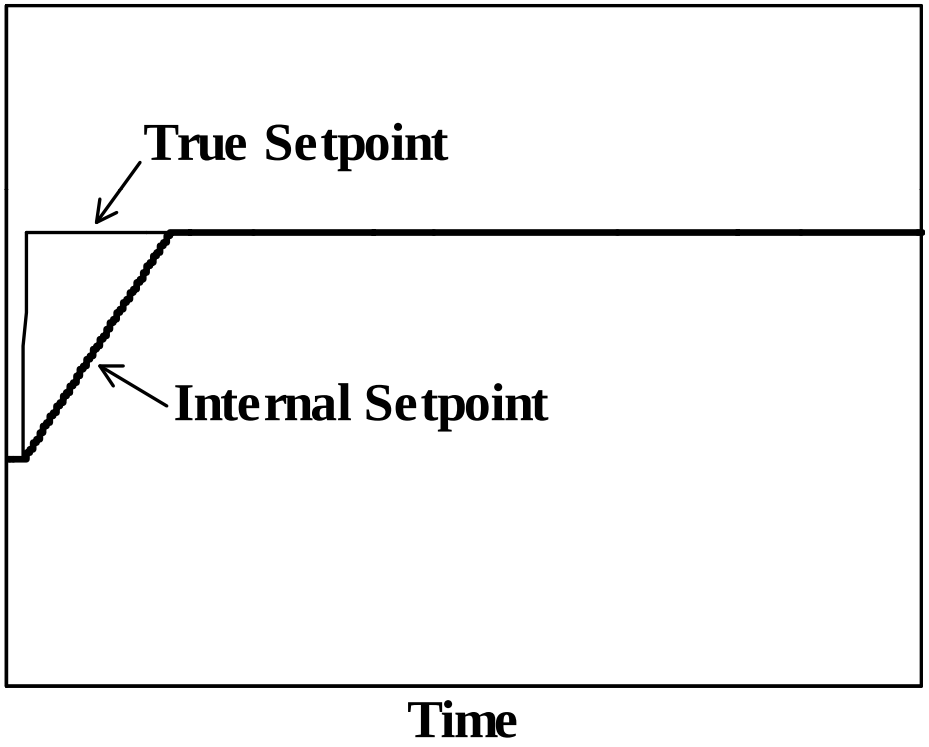
  Ini diperlukan untuk memastikan bahwa dua keluaran tersebut bertepatan dengan saat penggantian. Ini disebut ***bumpless transfer***

Bumpless Transfer (1)

- Dengan bumpless transfer, sebuah setpoint internal digunakan pada pengontrol dan setpoint internal dilandaikan pada laju yang rendah dari kondisi awal menjadi setpoint yang diinginkan untuk menyediakan startup yang lancar dari kontrol loop

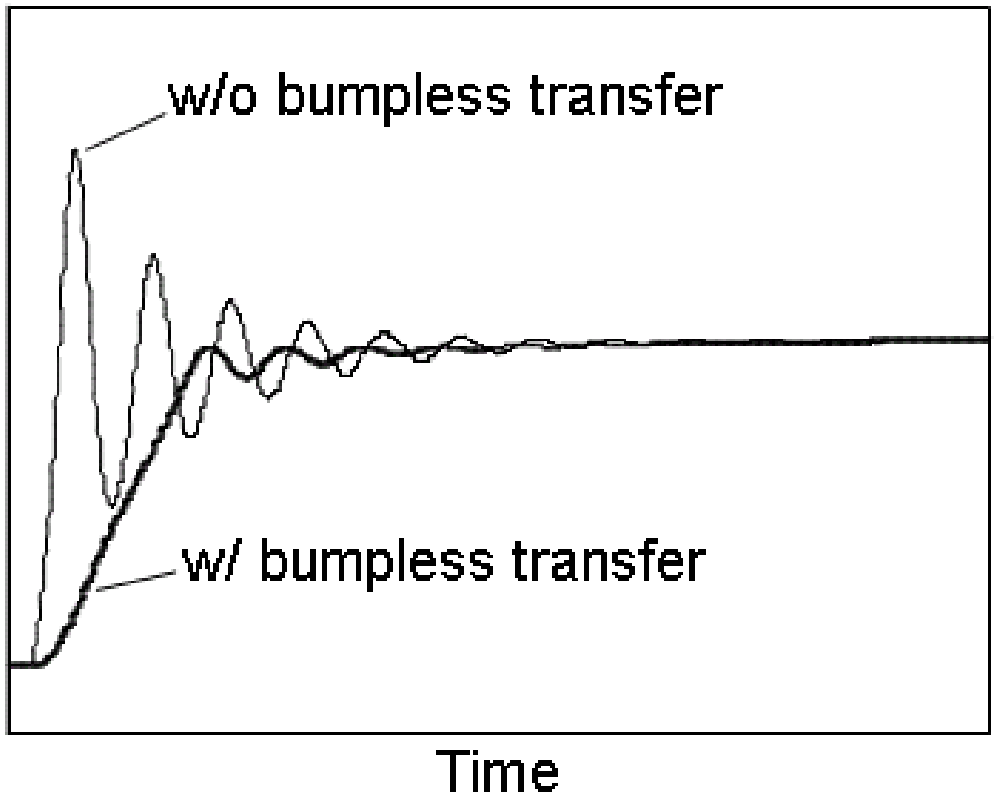
Bumpless Transfer (2)

- Perbandingan dari setpoint yang benar dan internal



Bumpless Transfer (3)

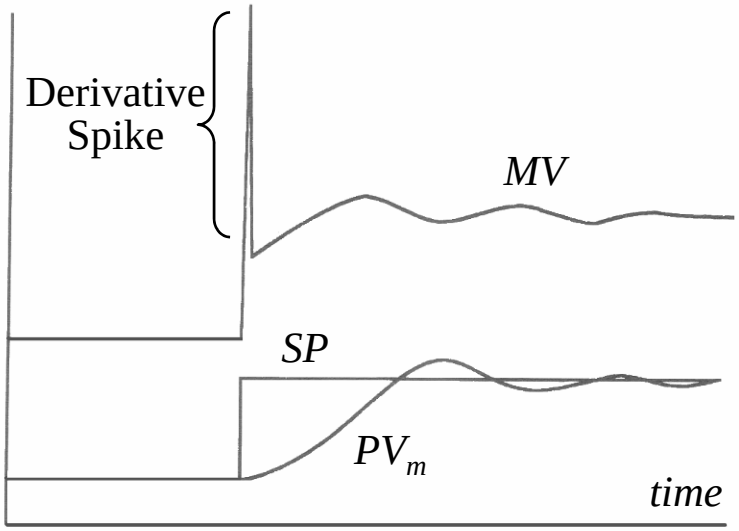
- Performance kontrol dengan atau tanpa bumpless transfer



Derivatif pada Proses Daripada Error (1)

Faktanya:

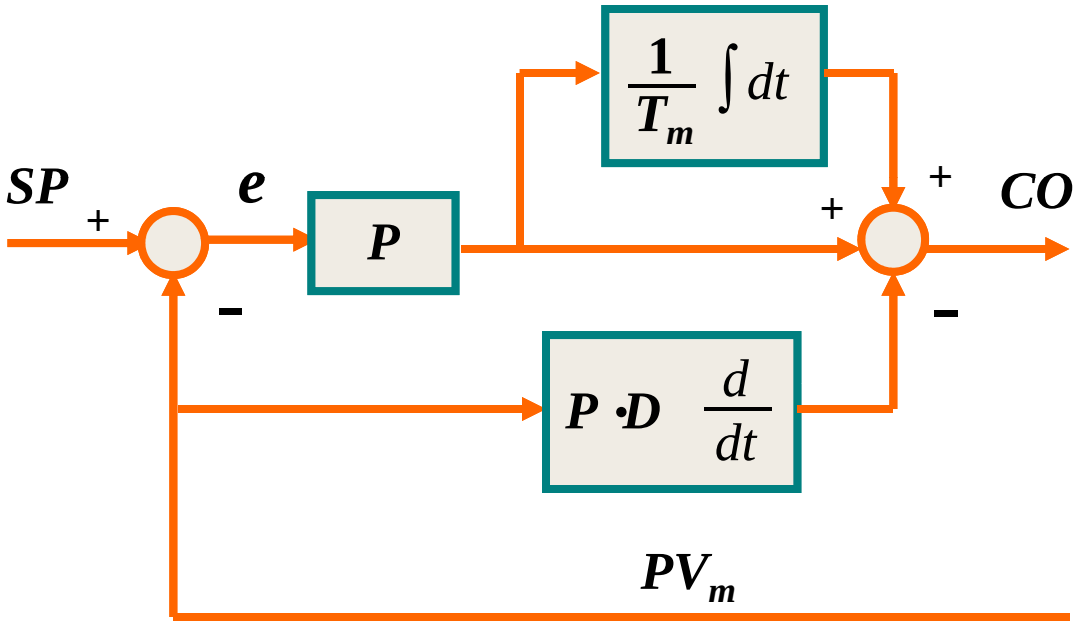
- Sebuah perubahan langkah pada set point menyebabkan perubahan langkah pada proses
- Derivatif beraksi pada tingkat perubahan dari error
- Tingkat perubahan dari perubahan langkah adalah sangat besar
- Perubahan langkah operator dari setpoint akan menyebabkan sebuah perubahan yang sangat besar pada keluaran, dan mengganggu proses



Gbr. Variabel proses dan respon katup terhadap perubahan setpoint menggunakan PID standar

Derivatif pada Proses Daripada Error (2)

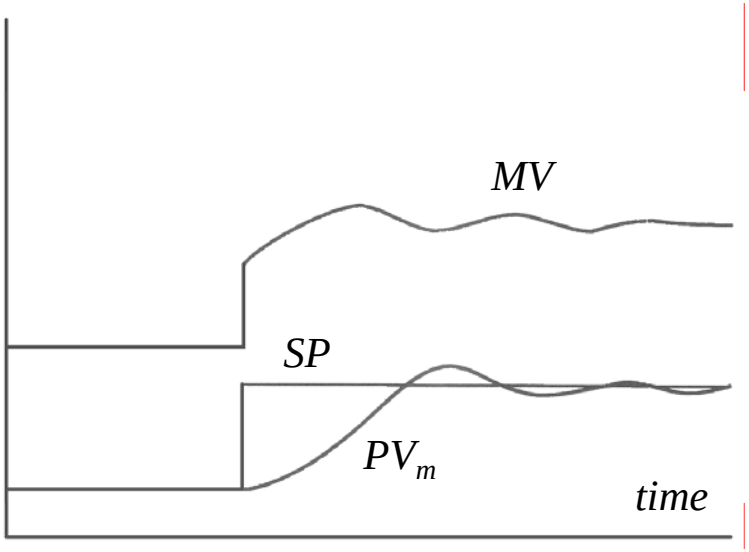
Solusi: Biarkan derivatif beraksi hanya pada **proses** daripada error



$$CO = P\left(e + \frac{1}{T_m}\int e\right) - P \cdot D\,\frac{dPV_m}{dt}$$

Derivatif pada Proses Daripada Error (3)

- Variabel proses dan respon katup terhadap perubahan setpoint



Gbr. Variabel proses dan respon katup terhadap perubahan setpoint menggunakan "**Derivative on Process Measurement**" PID

Derivatif pada Filtered Process Daripada Proses (1)

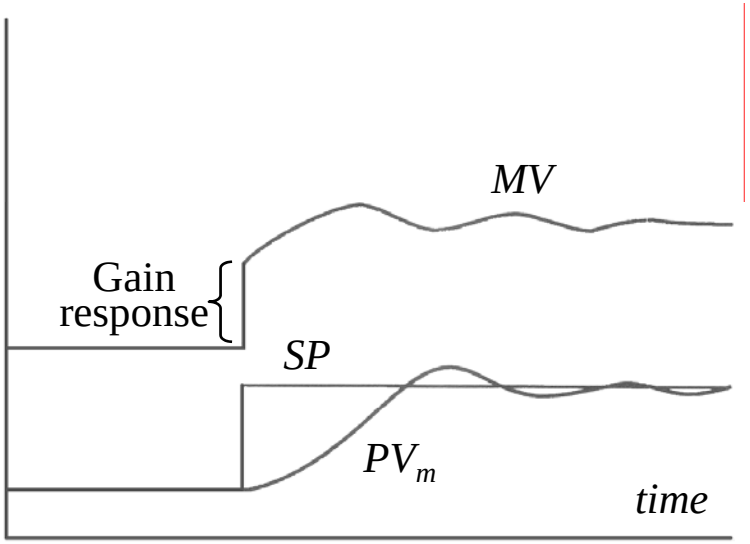
Faktanya:

- Mode derivatif hipersensitif terhadap noise

Gain pada Proses Daripada Error (1)

Faktanya:

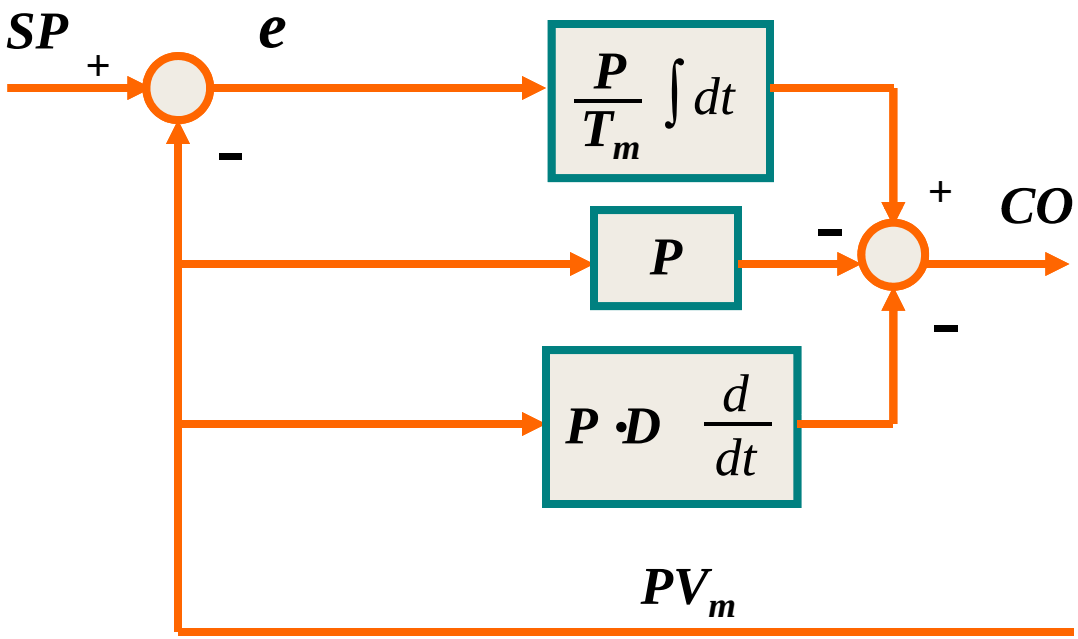
- Dalam aplikasi dengan gain yang tinggi, sebuah perubahan langkah bisa menyebabkan gerakan yang mendadak dan besar dari katup
- Tidak sehebat efek derivative, tetapi masih bisa mengganggu proses



Gbr. Variabel proses dan respon katup
Terhadap perubahan setpoint menggunakan
"**Gain on error and Derivative on Process Measurement**" PID

Gain pada Proses Daripada Error (2)

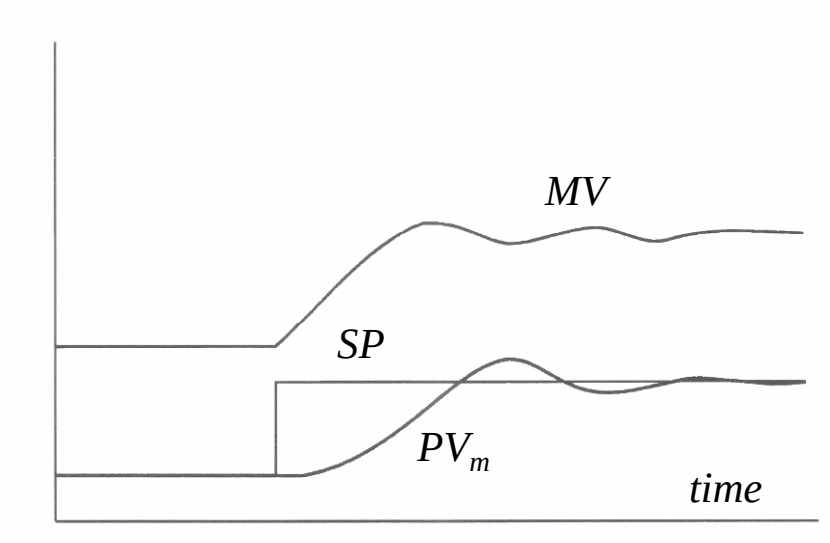
Solusi: Biarkan gain beraksi hanya pada proses daripada error



$$CO = \frac{P}{T_m}\int e\,dt + P\left(PV_m + \frac{dPV_m}{dt}\right)$$

Gain pada Proses Daripada Error (3)

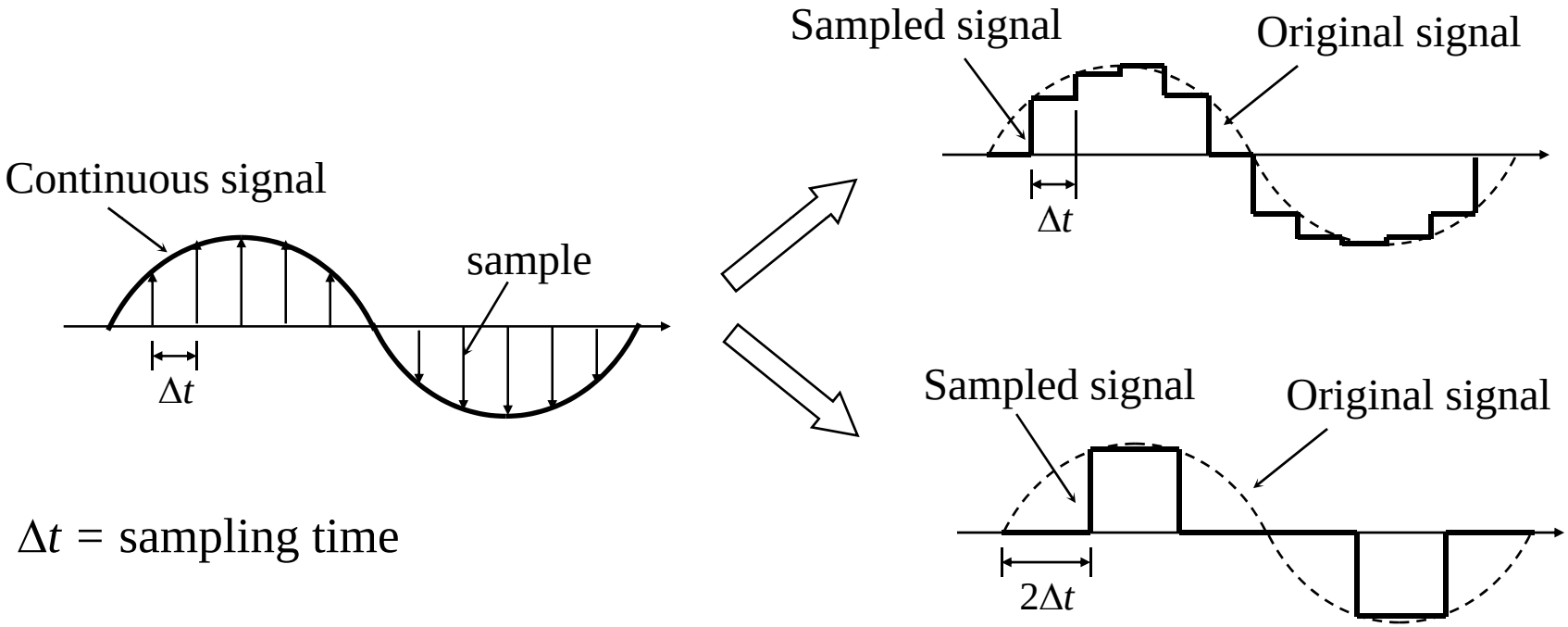
- Variabel proses dan respon katup ke sebuah perubahan setpoint



Gbr. Variabel proses dan respon katup pada perubahan setpoint menggunakan "**Gain and Derivative on Process Measurement**" PID

Algoritma PID Digital (1)

- Konsep akuisisi data



Δt = sampling time

Teorema sampling Shannon:

Frekuensi sampling harus lebih besar atau sama dengan dua kali frekuensi tertinggi yang muncul dalam sinyal yang akan diambil sample-nya

Algoritma PID Digital (2)

- Integral

$$\int e\,dt \cong \Delta t \cdot \Sigma\, e_i$$

- Derivative

$$\frac{de}{dt} \cong \frac{e_i - e_{i-1}}{\Delta t}$$

Formula Kontinyu:

$$CO = P\left(e + \frac{1}{T_m}\int e\,dt + D\frac{de}{dt}\right)$$

Formula Digital:

$$CO = P\left[e_i + \frac{\Delta t}{T_m}\sum e_i + \frac{D}{\Delta t}(e_i - e_{i-1})\right]$$

i = sampling instant

Algoritma PID Digital (3)

- Dua formula algoritma PID digital :
  - Formula posisional

$$CO_i = P\left[e_i + \frac{\Delta t}{T_m} \sum e_i \, dt + \frac{D}{\Delta t} (e_i - e_{i-1}) \right]$$

  - Formula kecepatan
    - Sudah menjadi sifatnya mempunyai fitur anti reset windup

$$CO_i = CO_{i-1} + P\left[(e_i - e_{i-1}) + \frac{\Delta t}{T_m} e_i + \frac{D}{\Delta t} (e_i - 2e_{i-1} + e_{i-2}) \right]$$

Beberapa Algoritma PID yang Ditawarkan (1)

Distributed Control System (DCS)

- Honeywell TDC 3000
  - Menawarkan 4 (empat) persamaan PID; A, B, C, dan D

TDC 3000	Mode P	Mode I	Mode D
Algoritma A	Error	Error	Error
Algoritma B	Error	Error	Pengukuran
Algoritma C	Pengukuran	Error	Pengukuran
Algoritma D	Tidak digunakan	Error	Tidak digunakan

Beberapa Algoritma PID yang Ditawarkan (2)

Distributed Control System (DCS)

- Foxboro I/A Series
- Yokogawa Centum CS 3000
- Bailey
- ABB

Beberapa Algoritma PID yang Ditawarkan (3)

Programmable Logic pengontrol (PLC)

- Modicon
- Allan-Bradley
  - PLC-5
  - SLC500
- Siemens
- Fuji Electric

Garis Pedoman untuk Loop Kontrol Umum (1)

Kontrol tekanan aliran dan cairan

- Respon cepat dengan tidak ada waktu penundaan (tidak ada pipa/transportasi)
- Biasanya dengan noise berfrekuensi-tinggi yang kecil
- Pengontrol PI dengan gain pengontrol menengah

Kontrol tingkat cairan

- Berisik dikarenakan deburan dan turbulensi
- Perolehan yang tinggi, aksi integral yang rendah dari pengontrol PI untuk integrating process
- Pengaturan konservatif untuk kontrol perataan ketika digunakan untuk mengurangi fluktuasi dari inlet stream

Garis Pedoman untuk Loop Kontrol Umum (2)

Kontrol Tekanan Gas

- Biasanya cepat dan melakukan tuning dengan sendirinya
- pengontrol PI dengan aksi integral kecil (waktu reset besar)

Kontrol Temperatur

- Berbagai jenis dari process nature
- Biasanya respon lambat dengan waktu penundaan
- Gunakan pengontrol PID untuk mempercepat respon

Garis Pedoman untuk Loop Kontrol Umum (3)

Kontrol Komposisi

- Serupa dengan kontrol suhu biasanya dengan noise yang lebih besar dan lebih banyak waktu penundaan
- Efektifvitas dari aksi derivatif adalah terbatas
- Kontrol suhu dan komposisi adalah kandidat utama untuk strategi kontrol yang lebih maju dikarenakan pentingnya dan sulitnya kontrol

Ringkasan Sesi

- Kontrol PID, yang merupakan algoritma kontrol paling banyak digunakan dalam aplikasi process kontrol, muncul dalam berbagai bentuk dan istilah
- Setiap istilah dari persamaan PID harus dipahami untuk mendapatkan sebuah kombinasi yang benar dari elemen kontrol PID untuk berbagai tujuan aplikasi kontrol proses