PID control of dead-time processes: robustness, dead-time compensation and constraints handling

Prof. Julio Elias Normey-Rico

Automation and Systems Department
Federal University of Santa Catarina

IFAC PID 18 – Ghent – May 2018
Dead-time processes are common in industry and other areas

Main dead-time (or delay) causes are:

- Transportation dead time (mass, energy)
- Apparent dead time (cascade of low order processes)
- Communication or processing dead time
• Dead time makes closed-loop control difficult
• Simplest solution:
  • PID - trade-off robustness and performance
• Basic dead-time compensator - Smith Predictor (SP)
• Improved solutions: Modified SP (ex. FSP)
• Advanced solution: Model Predictive Control - MPC

Most used in industry PID – DTC – MPC *

Industry 4.0 – complex controllers at low level

* A Survey on Industry Impact and Challenges Thereof. IEEE CONTROL SYSTEMS MAGAZINE 17
When to use advanced control?

Objectives: Analysis of PID, DTC and MPC for dead-time processes

PID: simple process models

DTC: compensates dead-time and can use high order models

MPC: is optimal and consider constraints

DEAD-TIME PROCESSES
1. Motivating examples, PID and DTC control.
2. Ideal control of dead-time processes
3. PID tuning using DTC ideas
   1. Unified tuning using FSP (stable and unstable plants)
   2. Trade-off performance-robustness
   3. Comparing PID and DTC
4. MPC, FSP and PID controllers
   1. Unconstrained case
   2. Constrained case – Using anti-windup
5. Conclusions
Motivating examples
Simple model with large delay and large modelling error

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

Even for a dominant delay process, PID offers a good response.

Simple model – big delay

Robust PID and DTC tuning (slow response)

Control

PID

DTC
**Fast response – small delay**

**Simple model with small modelling error**

\[ P_n(s) = \frac{e^{-0.2s}}{s+1} \]

**Well known delay (network)**

**Disturbance**

**Process**

**Control**

- **Fast PID tuning** (without oscillations)
- Even for a small delay DTC offers better response

**Robust DTC for the assumed modelling error**
To study the advantages of advanced controllers for dead-time processes related to:

- **Process dead-time**
- **Process modeling error (robustness)**
- **Other aspects:** Model complexity

**Constraints handling**
Ideal control of dead-time processes
Smith predictor of a pure delay process

\[ P(s) = e^{-Ls} \]

\[ H_{yr}(s) = \frac{Y(s)}{R(s)} = \frac{C(s)G_n(s)e^{-Ls}}{1 + C(s)G_n(s)} \]

\[ H_{yq}(s) = \frac{Y(s)}{Q(s)} = P_n(s)[1 - H_{yr}(s)] \]

\[ G_n(s) = 1 \quad C(s) = K_c \]

Ideal case \( K_c \rightarrow \infty \)

\[ H_{yr}(s) = e^{-Ls} \]

\[ H_{yq}(s) = e^{-Ls}\left[1 - e^{-Ls}\right] \]
Smith predictor of a FOPDT process

\[ P(s) = \frac{K_e}{1+sT} e^{-Ls} \]

Using \( C(s) = K_c \) and ideal case \( K_c \to \infty \)

\[ H_{yr}(s) = e^{-Ls} \]

\[ H_{yq}(s) = \frac{K_e}{1+sT} e^{-Ls} \left[ 1 - e^{-Ls} \right] \]

Open loop

SP: Only stable plants and slow responses

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
Normal index

\[ e(t) = r(t) - y(t) \]

\[ J = \int_{0}^{\infty} |e(t)| \, dt \]

**No controller can act before**

Process Output

2L

L

td
Ideal Control – Achievable Performance

Normal index

\[ J = \int_0^\infty |e(t)| \, dt \]

No controller can act before

\[ J_{sp} = \int_L^\infty |e(t)| \, dt \]

\[ J_{dr} = \int_{td+2L}^\infty |e(t)| \, dt \]

To compare controllers’ performance
Ideal Control – Achievable Performance

\[ e(t) = r(t) - y(t) \]

\[ J = \int_{0}^{\infty} |e(t)| \, dt \]

Normal index

\[ J_{sp} = \int_{L}^{\infty} |e(t)| \, dt \]

\[ J_{dr} = \int_{t_{d} + 2L}^{\infty} |e(t)| \, dt \]

SETPOINT

DISTURBANCE

No controller can act before

To compare controllers’ performance

“Ideal” \[ J_{\min} = 0 \]
Is it ideally possible to achieve $J_{\min} = 0$?

$
\begin{array}{c}
R(s) \\
\downarrow \ F(s) \\
\downarrow \ C(s) \\
\downarrow \ P(s) \\
\downarrow \ G_n(s) \\
\downarrow \ e^{-Ls} \\
\downarrow \ Y_p(s) \\
\downarrow \ F_r(s) \\
\downarrow \ Y(s) \\
\end{array}
$

Filtered Smith Predictor
Is it ideally possible to achieve $J_{\text{min}} = 0$?

The same $H_{yr}$ as SP

$$H_{yr}(s) = \frac{C(s)G_n(s)e^{-Ls}}{1 + C(s)G_n(s)}F(s)$$
Is it ideally possible to achieve $J_{min} = 0$?

The filter $F_r(s)$ allows:

- Eliminates the open-loop dynamics from the input disturbance response
- FSP for unstable plants
- FSP for ramp and other disturbances
- Robustness-Performance trade-off

The same $H_{yr}$ as SP

$$H_{yr}(s) = \frac{C(s)G_n(s)e^{-Ls}}{1 + C(s)G_n(s)}F(s)$$

$$H_{yq}(s) = P_n(s)\left[1 - H_{yr}(s)F_r(s)\right]$$
Is it ideally possible to achieve $J_{min} = 0$?

The filter $F_r(s)$ allows:

- Eliminates the open-loop dynamics from the input disturbance response
- FSP for unstable plants
- FSP for ramp and other disturbances
- Robustness-Performance trade-off

The same $H_{yr}$ as SP

$$H_{yr}(s) = \frac{C(s)G_n(s)e^{-Ls}}{1 + C(s)G_n(s)}F(s)$$

$$P(s) = \frac{K_e}{1+Ts}e^{-Ls}$$

$$H_{yr}(s) = e^{-Ls}$$

$$H_{yq}(s) = e^{-Ls} \left[ 1 - e^{-Ls} \right]$$

$J_{min} = 0!$
Example: Integrative plant

Simple Process \( P(s) = \frac{e^{-Ls}}{s} \)

Controller: \( C(s) = k_c \)

Filter \( F_r(s) = 1 + Ls \)

Ideal Tuning: \( k_c \to \infty \)

\[
H_{yr}(s) = e^{-Ls}
\]

\[
H_{yq}(s) = \frac{e^{-Ls}}{s} - \frac{e^{-2Ls}}{s} - Le^{-2Ls}
\]
PID design using FSP

Many FSP successful applications in practice:*

Termo-solar systems, Compression systems, Neonatal Care Unit.

FSP autotuning for simple process**

**Idea:** To derive a PID tuning for dead-time processes using the FSP approach

PID is a low frequency approximation of the FSP.

\[ C(s) = \frac{K_c (1+sT_i)(1+sT_d)}{sT_i (1+s\alpha T_d)} \]

*Torrico, Cavalcante, Braga, Normey-Rico, Albuquerque, I&EC Res. 2013

*Flesch, Normey-Rico, Control Eng. Practice, 2017
*Roca, Guzman, Normey-Rico, Berenguel, Yebra, Solar Energy, 2011
PID tuning using FSP
Tuning procedure

• Process models: FOPDT, IPDT, UFOPDT

\[ G_n(s) = \frac{K_p}{1+sT} \quad G_n(s) = \frac{K_p}{s} \quad G_n(s) = \frac{K_p}{sT-1} \]

• PI primary controller (only P for the IPDT)

\[ C(s) = K \frac{1+sT_i}{sT_i} \]

• FO predictor filter

\[ F_r(s) = \frac{1+sT_1}{1+sT_2} \] (tuning for step disturbances)

• Tuning for a delay-free-closed-loop system with pole (double pole) in \( s = -1/T_0 \)

\[ T_0 \] is the only tuning parameter for a trade-off robustness-performance
Equivalent 2DOF controller

\[ r(t) \rightarrow Feq(s) + e(t) \rightarrow Ceq(s) \rightarrow C_eq(s) \rightarrow P(s) \rightarrow y(t) \]

\[ C_{eq}(s) = \frac{C(s)F_r(s)}{1 + C(s)G_n(s)(1 - e^{-L_n s}F_r(s))}, \quad Feq(s) = \frac{F(s)}{F_r(s)} , \]

\[ e^{-L_n s} \rightarrow \frac{1 - 0.5L_n s}{1 + 0.5L_n s} \]

- \( C_{eq} \) avoids pole-zero cancellation
- \( T_o \) free tuning parameter
Tuning advantages of the predictor-PID

- Unified approach for FOPDT, IPDT and UFOPDT (L<2T)
- It has only one tuning parameter $T_0^*$
- Has similar performance than well known methods*
- It is a low frequency approximation of the ideal solution for first order dead-time models

* Astrom and Hagglund, Research Triangle Park, 2006
Performance Index

\[ J = \lambda \int_{t=t_s+L}^{t_d} |r(t) - y(t)| + (1 - \lambda) \int_{t=t_d+2L}^{t_{ss}} |r(t) - y(t)| \]

\[ \lambda \in [0, 1] \quad \lambda = 0.5 \text{ in this work} \]
Robustness

\[ P(j\omega) = P_n(j\omega)[1 + dP(j\omega)] \]

\[ C_{eq}(s) \text{ stabilizes } P_n(s) \]

\[ R(\omega) := \frac{|1 + C_{eq}(j\omega)P_n(j\omega)|}{|C_{eq}(j\omega)P_n(j\omega)|} \]

Robust condition \( R(\omega) > \overline{dP}(\omega) \geq |dP(j\omega)| \quad \forall \omega > 0 \)

Conservatism can be avoided separating dead-time uncertainties*

*Larsson and Hagglund (2009), ECC 2008
FSP-PID comparative analysis

Stable Lag dominant $T=5L$

$R_{FSP}$  $T_0$  $R_{PID}$

modelling error

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
FSP-PID comparative analysis

Stable Lag dominant $T = 5L$

Stable Delay dominant $L = 5T$

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
FSP-PID comparative analysis

- Robust tuning $J_{FSP} \approx J_{PID}$
- Fast tuning $J_{FSP} < J_{PID}$
FSP-PID comparative analysis

- Stable Lag dominant $T = 5L$
  - Robust tuning $J_{FSP} \approx J_{PID}$
- Stable Delay dominant $L = 5T$
  - Fast tuning $J_{FSP} < J_{PID}$

PID for robust solutions
FSP has advantages with good models
Motivation examples  
Ideal control  
PID tuning  
MPC FSP and PID  
Conclusions

FSP-PID comparative analysis

- Similar to Lag-dominant plants

Integrative plant

Unstable plant

- Same conclusions as in FOPDT

- UFOPDT Robustness has a limit increasing $T_0$ *

* Normey-Rico and Camacho, 2007, Springer
FSP-PID comparative analysis

Tuning: Trade-off Robustness-Performance

- Minimise $J$ for robust stability for a given modelling error

**Particular tuning using:**

$$R(\omega) > \overline{dP}(\omega) \quad \forall \omega > 0$$

- Minimise $J$ for robust stability for a given

**General tuning using**

$$R_m = \min_{\omega} R(\omega)$$

* Grimholt and Skogestad 2012, IFAC PID 2012.
Tuning: Trade-off Robustness-Performance

- Minimise $J$ for robust stability for a given modelling error

**Particular tuning using:** $R(\omega) > \overline{dP}(\omega)$ $\forall \omega > 0$

- Minimise $J$ for robust stability for a given $R_m = \min_{\omega} R(\omega)$

**General tuning using** $R_m$ (or $M_S$)

Control effort (total variation) and noise attenuation are directly related to robustness indexes as $R_m$ (or $M_s$)*

---

* Grimholt and Skogestad 2012, IFAC PID 2012.
Conclusions

- **Case 1: poor model information (large modelling error)**
  - Simple model is used for tuning
  - High robustness is mandatory
  - Step disturbances

  **PID will be the best solution, even for dead-time dominant systems**

- **Case 2: good model is available (small modelling error)**
  - Fast responses are required
  - Low robustness is enough
  - Complex models or disturbances

  **FSP will be better (even for lag-dominant systems) because of the PID nominal limitations**
Conclusions

Concerning dead-time: dead-time value is less important than dead-time modelling error.

Implementation issues:

• FSP is implemented as a 2DOF discrete controller
• FSP is a complex algorithm (delay order (in samples) + model order)
• PID is simple to implement

General problems in industry: large modelling error, noise, simple models and solutions

Use a well tuned PID for dead-time processes
Example 1: High-order system

\[ P(s) = \frac{e^{-s}}{(s+1)^3} \]

\[ P_n(s) = \frac{e^{-2s}}{(2s+1)} \]

Robust tuning for \( Ms=1.2 \)

Prediction Model for FSP

PID tuning using SWORD * tool

FSP and PID have the same performance


**SWORD Matlab software tool.**
Example 2: PID, SP and FSP

\[ P(s) = \frac{e^{-10s}}{1 + \frac{2\xi s}{\omega_n} + \frac{s^2}{\omega_n^2}} \]

\[ \xi = 0.2, \ \omega_n = 1 \]

- SP and FSP with the same primary PID controller
- PID tuning for min IAE for \( M_s = 2 \) (using sword tool)

Max. delay error 20%

\[ M_s = 2, \ 40\% \ better \]

Open-loop oscillatory disturbance response

Performance Analysis

\[ J = \int_0^\infty |e(t)| \, dt \]

FSP 14\% better

\[ J_{dr} = \int_{t_d+2L}^\infty |e(t)| \, dt \]

FSP 40\% better

Robustness: FSP stable up to -35\% or +35\% delay error, SP unstable for 20\% delay error
Example 2: PID, SP and FSP

\[ P(s) = \frac{e^{-10s}}{1 + \frac{2\xi s}{\omega_n} + \frac{s^2}{\omega_n^2}} \]

\[ \xi = 0.2, \ \omega_n = 1 \]

- SP unstable for this case
- PID and FSP similar responses
• In real process control action is limited, as well as slew rate

• Also, process output should be between limits

• Anti-windup (AW) can be used to mitigate the effect of the saturation in the integral action in PID and FSP

• MPC appears as a direct solution to implement optimal control under system constraints

When is MPC a better choice?
MPC, FSP and PID

GPC – Generalized predictive controller
General MPC idea

- Process
- Model
- Control Computation
  - Min J(u)
- Model output (future values)
- Constraints
- Reference
- Plant output
- Control action
GPC analysis for Dead-time Processes

General MPC idea

GPC cost

\[ J = \sum_{j=d+1}^{d+N_y} [y(k+j|k) - w(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda [\Delta u(k+j)]^2, \]

GPC Model

\[ A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{e(t)}{\Delta} \]

\[ L = dT_s \]
GPC analysis for Dead-time Processes

General MPC idea

GPC cost

\[ J = \sum_{j=d+1}^{d+N_y} [y(k+j|k) - w(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda [\Delta u(k+j)]^2, \]

GPC Model

\[ A(z^{-1})y(k) = z^{-d} B(z^{-1})u(k-1) + \frac{e(t)}{\Delta} \]

\[ L = dT_s \]
GPC analysis for Dead-time Processes

General MPC idea

\[
J = \sum_{j=d+1}^{d+N_y} [y(k+j|k) - w(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda [\Delta u(k+j)]^2,
\]

GPC cost

A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{e(t)}{\Delta}

L = dT_s

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
Motivation examples
Ideal control
MPC FSP and PID
Conclusions

GPC analysis for Dead-time Processes

General MPC idea

control action
Process
Plant output

control action
Model

Model output (future values)

Constraints
Reference

Control Computation
Min J(u)

TUNNING

GPC cost

\[ J = \sum_{j=d+1}^{d+N_y} [y(k+j|k) - w(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda [\Delta u(k+j)]^2, \]

Process
Model
Reference
Constraints

\[ A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{e(t)}{\Delta} \quad L = dT_s \]

\[ y(k + d + j/k) = f(u_{fut}, y_{past}, u_{past}) \]

\[ J = J(u_{fut}) \]
GPC analysis for Dead-time Processes

Prediction computation

\[ y(k + j/k) \quad j = 1 \ldots d \quad \text{past} \]

\[ y(k + d + j/k) \quad j = 1 \ldots N \quad \text{Delay horizon} \]

\[ u_{past} \quad k \]

\[ y_{past} \quad k+d \]

\[ k+d+1 \]

\[ k+d+N \]

\[ \text{Prediction horizon of } J \]
GPC analysis for Dead-time Processes

Prediction computation

\[ y(k + j/k) \quad j = 1...d \quad \longleftrightarrow \quad y(k + d + j/k) \quad j = 1...N \]

GPC structure?
GPC analysis for Dead-time Processes

Prediction computation

\[ y(k + j/k) \quad j = 1...d \]
\[ y(k + d + j/k) \quad j = 1...N \]

GPC structure? (unconstrained)

\[ \begin{align*}
  & \text{u}_{past} \\
  & \text{y}_{past}
\end{align*} \]

\[ \begin{align*}
  & \text{Delay horizon} \\
  & \text{Prediction horizon of } J
\end{align*} \]

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
GPC analysis for Dead-time Processes

Prediction computation

\[ y(k + j/k) \quad j = 1 \ldots d \quad \text{versus} \quad y(k + d + j/k) \quad j = 1 \ldots N \]

\[ u_{\text{past}} \quad \text{past} \quad k \quad k+d \quad k+d+1 \quad \text{Prediction horizon of } J \]

GPC structure? (unconstrained)

\[ C(z) \text{ integral action} \]

order \{G_n(z)\} \rightarrow \text{order } \{C(z), F_r(z)\}

coefficients related to \( N, N_u, \lambda \)
Unconstrained GPC structure

- GPC is equivalent to a discrete FSP
- FSP can be tuned using GPC method (exactly the same solution)
- FSP-MPC can be used (for robust controllers and easy tuning)*
- For 1st order models $\rightarrow$ GPC $\Rightarrow$ 2DOF FSP (PI primary controller)

Comparison FSP-PID is valid for GPC-PID for 1st order models

Is valid for other linear MPC (simply a model rearrangement)

Constrained case?

* Normey-Rico and Camacho, 2007, Springer
* Lima, Santos and Normey-Rico, 2015, ISA Transactions
Constrained GPC

\[
\begin{align*}
\underline{U} & \leq u(k) \leq \overline{U} \quad \forall k \geq 0, \\
u & \leq u(k) - u(k-1) \leq \overline{u} \quad \forall t \geq 0, \\
y & \leq y(k) \leq \overline{y} \quad \forall k \geq 0.
\end{align*}
\]

\[
u = [\Delta u(k) \ldots \Delta u(k + N_u - 1)]
\]
Constrained GPC

\[ \frac{U}{\underline{u}} \leq u(k) \leq \frac{\bar{U}}{\bar{u}} \quad \forall k \geq 0, \]
\[ u \leq u(k) - u(k-1) \leq \bar{u} \quad \forall t \geq 0, \]
\[ y \leq y(k) \leq \bar{y} \quad \forall k \geq 0. \]

\[ u = [\Delta u(k) \ldots \Delta u(k + N_u - 1)] \]
**Constrained GPC**

\[ U \leq u(k) \leq \bar{U} \quad \forall k \geq 0, \]
\[ u \leq u(k) - u(k-1) \leq \bar{u} \quad \forall t \geq 0, \]
\[ y \leq y(k) \leq \bar{y} \quad \forall k \geq 0. \]

\[
\begin{align*}
\min_u & \quad \frac{1}{2} u^T H u + b^T u + f_0, \\
\text{s. t.} & \quad Ru \leq r
\end{align*}
\]

All constraints are written as a linear inequality on \( u \)

\[ u = [\Delta u(k) \ldots \Delta u(k + N_u - 1)] \]
Constrained GPC

\[
\begin{align*}
\frac{U}{2} \text{ min } & \quad \frac{1}{2} u^T H u + b^T u + f_0, \\
\text{s. t. } & \quad R u \leq r
\end{align*}
\]

- All constraints are written as a linear inequality on \( u \)

\[
\begin{align*}
u(k) \leq u(k) \leq U, & \quad \forall k \geq 0, \\
u(k) \leq u(k) - u(k-1) \leq u, & \quad \forall t \geq 0, \\
y(k) \leq y(k) \leq \bar{y}, & \quad \forall k \geq 0.
\end{align*}
\]

\[
u = [\Delta u(k) \ldots \Delta u(k + N_u - 1)]
\]

- QP solved at each sample time
- Only \( u(k) \) is applied
- The horizon window is displaced
**Constrained GPC**

\[
\begin{align*}
\underline{U} & \leq u(k) \leq \overline{U} \quad \forall k \geq 0, \\
u & \leq u(k) - u(k-1) \leq \overline{u} \quad \forall t \geq 0, \\
y & \leq y(k) \leq \overline{y} \quad \forall k \geq 0.
\end{align*}
\]

\[
\min_{u} \frac{1}{2} u^T H u + b^T u + f_0, \\
s. t. \\
Ru \leq r
\]

All constraints are written as a linear inequality on u

\[
u = [\Delta u(k) \ldots \Delta u(k + N_u - 1)]
\]

- QP solved at each sample time
- Only \( u(k) \) is applied
- The horizon window is displaced

GPC gives goods results with small \( N_u \) (in many applications \( N_u=1 \) is enough*)

* De Keyser and Ionescu, IEEE CCA 2003
AW for FSP and PID

AW scheme

\[ u(k) = u_i(k) + u_d(k) \]

- \( u_i(k) \) has the integral action of PID or FSP
- \( u_d(k) \) has the rest of the control action of PID or FSP

AW originally derived for control action constraints

Several AW strategies in literature
AWP with error recalculation (ER)

Recalculation of the error signal at every sample

Objective: to maintain the consistence between $u(k)$ (computed) and $u_r(k)$ (applied)

* Flesch and Normey-Rico, Control Eng. Practice, 2017
* Silva, Flesch and Normey-Rico, IFAC PID 18
AWP with error recalculation (ER)

Recalculation of the error signal at every sample
Objective: to maintain the consistence between \( u(k) \) (computed) and \( u_r(k) \) (applied)

PID case

\[
\begin{align*}
    u(k) &= u(k-1) + n_0 e(k) + n_1 e(k-1) + n_2 e(k-2) \\
    u(k) > u_{max} &\rightarrow u_r(k) = u_{max}
\end{align*}
\]

* Flesch and Normey-Rico, Control Eng. Practice, 2017
* Silva, Flesch and Normey-Rico, IFAC PID 18
Recalculation of the error signal at every sample

Objective: to maintain the consistence between $u(k)$ (computed) and $u_r(k)$ (applied)

\[ u(k) = u(k - 1) + n_0 e(k) + n_1 e(k - 1) + n_2 e(k - 2) \]

**PID case**

\[ u(k) > u_{max} \rightarrow u_r(k) = u_{max} \]

Consider:  
\[ u_r(k) = u(k - 1) + n_0 e^*(k) + n_1 e(k - 1) + n_2 e(k - 2) \]

* Flesch and Normey-Rico, Control Eng. Practice, 2017
* Silva, Flesch and Normey-Rico, IFAC PID 18
AWP with error recalculation (ER)

Recalculation of the error signal at every sample

Objective: to maintain the consistence between $u(k)$ (computed) and $u_r(k)$ (applied)

$u(k) = u(k - 1) + n_0 e(k) + n_1 e(k - 1) + n_2 e(k - 2)$

$u(k) > u_{max} \rightarrow u_r(k) = u_{max}$

Consider:

$u_r(k) = u(k - 1) + n_0 e^*(k) + n_1 e(k - 1) + n_2 e(k - 2)$

$e^*(k) = e(k) + \frac{u_r(k) - u(k)}{n_0}$

Used in the code to update the error:

$e(k-1) = e^*(k)$

* Flesch and Normey-Rico, Control Eng. Practice, 2017
* Silva, Flesch and Normey-Rico, IFAC PID 18
AWP with error recalculation (ER)

Recalculation of the error signal at every sample

Objective: to maintain the consistence between $u(k)$ (computed) and $u_r(k)$ (applied)

\[
\begin{align*}
\text{PID case} & \\
\quad u(k) &= u(k - 1) + n_0 e(k) + n_1 e(k - 1) + n_2 e(k - 2) \\
\quad u(k) &> u_{\text{max}} \rightarrow u_r(k) = u_{\text{max}}
\end{align*}
\]

Consider:

\[
\begin{align*}
\quad u_r(k) &= u(k - 1) + n_0 e^*(k) + n_1 e(k - 1) + n_2 e(k - 2) \\
\quad e^*(k) &= e(k) + \frac{u_r(k) - u(k)}{n_0}
\end{align*}
\]

\[\text{Used in the code to update the error: } e(k-1)=e^*(k)\]

ER* better results, principally in noise environment

* Flesch and Normey-Rico, Control Eng. Practice, 2017
* Silva, Flesch and Normey-Rico, IFAC PID 18
Including several constraints in AW scheme

\[ u(k) < U_{max} \quad \Delta u(k) < \Delta u_{max} \quad y(k) < y_{max} \]
Including several constraints in AW scheme

- $u(k) < U_{max}$
- $\Delta u(k) < \Delta u_{max}$
- $y(k) < y_{max}$
AW for dead-time processes

Including several constraints in AW scheme

\[ u(k) < U_{max} \]
\[ \Delta u(k) < \Delta u_{max} \]
\[ y(k) < y_{max} \]

Direct

\[ \Delta u(k) = u(k) - u(k - 1) < \Delta u_{max} \]

\[ u(k) < \Delta u_{max} + u(k - 1) \]
Including several constraints in AW scheme

Direct

\[ u(k) < U_{\text{max}} \]

\[ \Delta u(k) = u(k) - u(k - 1) < \Delta u_{\text{max}} \]

\[ u(k) < \Delta u_{\text{max}} + u(k - 1) \]

\[ \Delta u(k) < \Delta u_{\text{max}} \]

\[ y(k) < y_{\text{max}} \]

Using prediction ideas

\[ y(k + j) < y_{\text{max}} \quad \forall \quad j = 1 \ldots N_y \]

Predictions

\[ y(k - i) \]
\[ u(k - i) \]
\[ u(k + j) \]

\[ y(k + d + j/t) \]
AW for dead-time processes

Including several constraints in AW scheme

\[ u(k) < U_{max} \]
\[ \Delta u(k) < \Delta u_{max} \]
\[ y(k) < y_{max} \]

Direct

\[ \Delta u(k) = u(k) - u(k - 1) < \Delta u_{max} \]

\[ u(k) < \Delta u_{max} + u(k - 1) \]

Using prediction ideas

\[ y(k + j) < y_{max} \quad \forall \quad j = 1 \ldots N_y \]

Predictions

MODEL

\[ y(k - i) \]
\[ u(k - i) \]
\[ u(k + j) \]

Assuming \( N_u = 1 \)

\[ u(k + j) = u(k) \quad \forall j \]

\[ y(k + d + j/k) = f(u(k), y(k - i), u(k - i)) \]
SMALL CASE: FOPDT

\[ y(k) = ay(k - 1) + bu(k - d - 1) \]
SIMPLE CASE: FOPDT

\[ y(k) = ay(k - 1) + bu(k - d - 1) \]

\[ y(k + d) = a^d y(k) + ba^{d-1} u(k - d) + \ldots + bu(k - 1) \]
AW for dead-time processes

SIMPLE CASE: FOPDT

\[ y(k) = ay(k - 1) + bu(k - d - 1) \]

\[ y(k + d) = a^d y(k) + b a^{d-1} u(k - d) + ... + bu(k - 1) \]

\[ y(k + d + j) = a^j y(k + d) + (a^{j-1} + a^{j-2} + ... + 1)b \frac{u(k)}{K_j} \]
**Motivation examples**

**Ideal control**

**PID tuning**

**MPC FSP and PID**

**Conclusions**

**SIMPLE CASE: FOPDT**

\[ y(k) = ay(k-1) + bu(k - d - 1) \]

\[ y(k + d) = a^d y(k) + ba^{d-1}u(k - d) + \ldots + bu(k - 1) \]

\[ y(k + d + j) = a^j y(k + d) + (a^{j-1} + a^{j-2} + \ldots + 1)b \frac{u(k)}{K_j} \]

\[ y(k + d + j) < y_{max} \]

\[ u(k) < \frac{y_{max} - a^j y(k+d)}{K_j} \]
AW for dead-time processes

**SIMPLE CASE:** FOPDT

\[ y(k) = ay(k - 1) + bu(k - d - 1) \]

\[ y(k + d) = a^d y(k) + ba^{d-1}u(k - d) + ... + bu(k - 1) \]

\[ y(k + d + j) = a^j y(k + d) + (a^{j-1} + a^{j-2} + ... + 1)b \frac{u(k)}{K_j} \]

\[ y(k + d + j) < y_{max} \quad \Rightarrow \quad u(k) < \frac{y_{max} - a^j y(k+d)}{K_j} \]

\[ u(k) < \min\{U_{max}; \Delta u_{max} + u(k - 1); \frac{y_{max} - a^j y(k+d)}{K_j} \} \]
• Constrained GPC or FSP-ER-AW
  • Good tuned FSP with ER-AWP equivalent to GPC (Nu=1)
  • On-line optimization is avoided with FSP
  • FSP filter tuning is easy in practice

Several successful applications in solar systems and refrigeration plants *

• In robust industrial solutions \( \rightarrow \) PID-ER-AW
  • Simple models are used
  • Robust tuning (low \( M_s \) or high \( R_m \) values)

Experiments: Electrical water heater

\[ y = T_o - T_i \]

Normalized Control variable (number pulses)

\[ u_{max} = 1, \quad u_{min} = 0 \]

Model identification: step test

\[ P(s) = \frac{18.7e^{-8s}}{13.1s+1} \]

GPC - \( N = 60, N_u = 10, \lambda_n = 1 \)

PID - \( T_0 = 8 \)

Same IAE performance
PID smoother control action
Temperature control

**PID** - $T_0 = 8$

New GPC tuning to accelerate the responses

GPC - $N = 60$, $N_u = 10$, $\lambda_n = 0.3$

Problems:
- Small performance improvement
- Lower robustness
- Lower noise attenuation

PID is simpler and better

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
Motivation examples
Ideal control
PID tuning
MPC FSP and PID
Conclusions

\[
\frac{T(s)}{U(s)} = \frac{0.76}{304.7s+1}e^{-108s}
\]

\[u_{max} = 95\%\]

\[u_{min} = 5\%\]

Compressor-test system

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
Compressor-test system

$$\frac{T(s)}{U(s)} = \frac{0.76}{304.7s + 1} e^{-108s}$$

$$u_{\text{max}} = 95\%$$

$$u_{\text{min}} = 5\%$$

Important
- To maintain Inlet temperature
- Fast set-point response
- Fast disturbance rejection
- Delay error well estimated
\[ \frac{T(s)}{U(s)} = \frac{0.76}{304.7s+1} e^{-108s} \]

\[ u_{max} = 95\% \]

\[ u_{min} = 5\% \]

Compressor-test system

Important
- To maintain Inlet temperature
- Fast set-point response
- Fast disturbance rejection
- Delay error well estimated

FSP ER-AWP

PID control of dead-time processes: robustness, dead-time compensation and constraints handling
Compressor-test system

\[ \frac{T(s)}{U(s)} = \frac{0.76}{304.7s+1} e^{-108s} \]

\[ u_{max} = 95\% \]

\[ u_{min} = 5\% \]

Fast disturbances

Important
- To maintain Inlet temperature
- Fast set-point response
- Fast disturbance rejection
- Delay error well estimated

FSP ER-AWP
Conclusions
Conclusions

• When controlling dead-time processes....
  • Performance measurement after the dead-time
  • Ideal solution can be achieved by FSP (or other improved DTC)
  • Dead-time estimation error is very important
  • Constrained case: ER AW FSP can be equivalent to MPC

• PID for dead-time processes
  • Can be tuned as a low order approximation of FSP
  • Performance improvement is limited in complex cases
  • For high robust solutions PID is equivalent to FSP (even for high L)
  • ER AW PID sub-optimal solution with good results.
Conclusions

• When controlling dead-time processes....
  • Performance measurement after the dead-time
  • Ideal solution can be achieved by FSP (or other improved DTC)
  • Dead-time estimation error is very important
  • Constrained case: ER AW FSP can be equivalent to MPC

• PID for dead-time processes
  • Can be tuned as a low order approximation of FSP
  • Performance improvement is limited in complex cases
  • For high robust solutions PID is equivalent to FSP (even for high L)
  • ER AW PID sub-optimal solution with good results.

Low-order-process models
Large modelling error
Noise environment
Typical constraints

Well tuned robust PID with AW is the best option
Conclusions

• PID still has an important figure in process industry

• DTC strategies with PI or PID primary controllers can be considered as extensions of simple PID control and used in particular cases

• Improved AW PID algorithms (or FSP AW) can be the solution in modern real-time distributed control systems for simple constrained systems

• MPC solutions are important in complex well modeled systems and at second level control
Thanks!

For your attention

julio.normey@ufsc.br